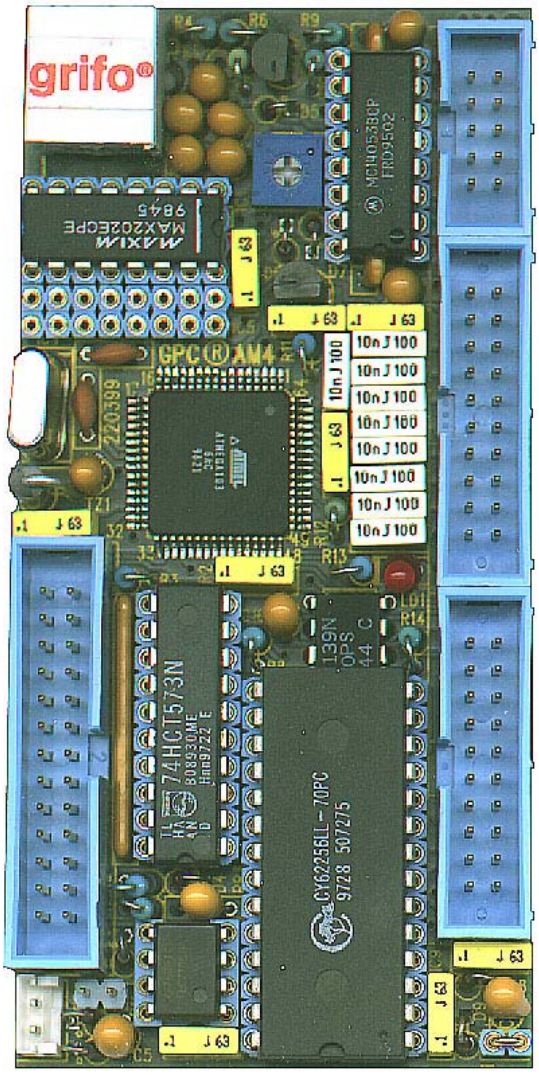
GPC® AM4

General Purpose Controller AVR Mega, serie 4

MANUALE TECNICO

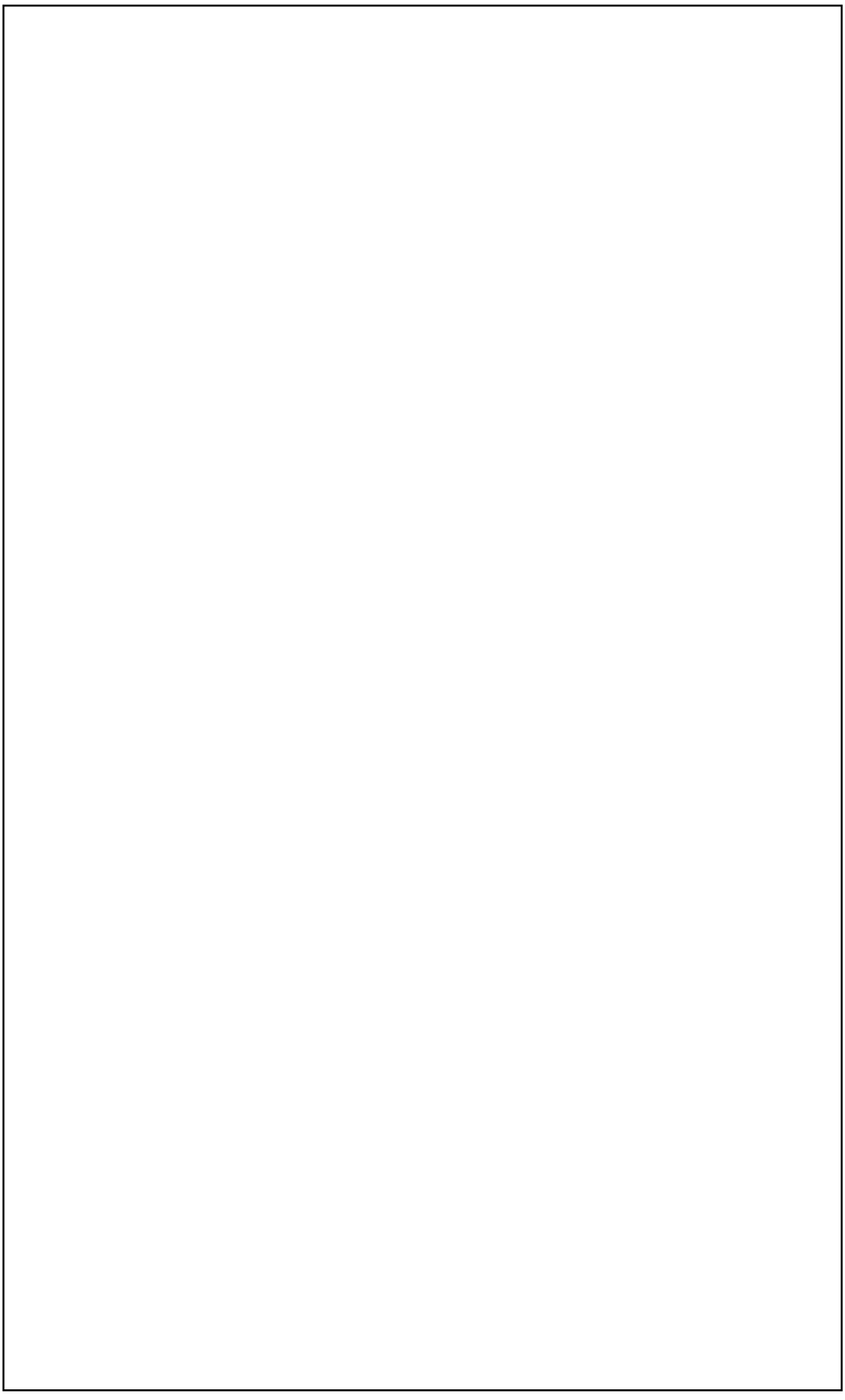


grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® AM4 Edizione 3.00 *Rel. 04 Aprile 2003*

abaco, **GPC®**, **grifo®**, *sono marchi registrati della ditta* **grifo®**



GPC® AM4

General Purpose Controller AVR Mega, serie 4

MANUALE TECNICO

Modulo Intelligente **Abaco® BLOCK**, della **serie 4**, nel formato 100x50 mm. Contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**. CPU **ATmega103** da **5,5296 MHz** con: **128K FLASH EPROM** interna; **4K RAM** interna; **4K EEPROM** interna; **8** linee di **A/D converter** da 10 bit, 0÷2.49 V, su connettore standard; **2 timer counter** ad 8 bit in grado di generare seganli **PWM**; **1 timer counter** a 16 bit in grado di generare due segnali **PWM** da 10 bit; **Watch Dog** interamente gestibile via software; **Comparatore** analogico; **15** sorgenti di **Interrupt** (interne ed esterne). **32K SRAM** esterna su zoccolo. **Real Time Clock** in grado di gestire giorno, mese, anno, giorno della settimana, ore, minuti, secondi e di emettere interrupt con cadenze definibili da software. Circuiteria di **back up** per SRAM esterna ed RTC, con batteria al **LITIO**. Fino a **20** linee di **I/O** TTL su connettori standard; **Jumpers** di configurazione acquisibile via software. 1 linea seriale in **RS232**, settabile in **RS422**, **RS485** o **current loop**. Interfaccia per **ABACO® I/O BUS** su connettore standard da 26 vie. Interfaccia **ISP** compatibile con programmatori **grifo® MP AVR-51+** e **SEP 40** o **UISP-S4** della EQUINOX. Possibilità di funzionamento in modalità **Idle** o **Power Down**. Unica alimentazione esterna da **5 Vdc**, **50 mA**, protetta tramite **TransZorb™**. Vasta disponibilità di software di sviluppo quali **Assembler**, **AVR Studio**, **DDS MICRO-C AVR**; **ICC AVR**; **BASCOM AVR**; ecc.

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® AM4 Edizione 3.00 *Rel. 04 Aprile 2003*

abaco, **GPC®**, **grifo®**, *sono marchi registrati della ditta* **grifo®**

Vincoli sulla documentazione grifo® Tutti i Diritti Riservati

Nessuna parte del presente manuale può essere riprodotta, trasmessa, trascritta, memorizzata in un archivio o tradotta in altre lingue, con qualunque forma o mezzo, sia esso elettronico, meccanico, magnetico ottico, chimico, manuale, senza il permesso scritto della **grifo®**.

IMPORTANTE

Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo®** non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo®**.
Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:

Attenzione: Pericolo generico

Attenzione: Pericolo di alta tensione

Marchi Registrati

abaco , **GPC®**, **grifo®** : sono marchi registrati della **grifo®**.
Altre marche o nomi di prodotti sono marchi registrati dei rispettivi proprietari.

INDICE GENERALE

INTRODUZIONE 1

VERSIONE SCHEDA 1

INFORMAZIONI GENERALI 2
- ALIMENTAZIONE 3
- COMUNICAZIONE SERIALE 3
- CLOCK 3
- PROCESSORE DI BORDO 4
- ABACO® I/O BUS 4
- MEMORIE 6
- REAL TIME CLOCK 6
- WATCH DOG 6
- A/D CONVERTER 6
- LINEE DI I/O DIGITALI 7
- TIMER COUNTER 7
- INTERFACCIA ISP 7

SPECIFICHE TECNICHE 8
- CARATTERISTICHE GENERALI 8
- CARATTERISTICHE FISICHE 8
- CARATTERISTICHE ELETTRICHE 9

INSTALLAZIONE 10
- CONNESSIONI 10
  - CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP 10
  - CN1 - CONNETTORE PER ABACO® I/O BUS 11
  - CN3 - CONNETTORE PER LINEA SERIALE 12
  - CN5 - CONNETTORE PER I/O DIGITALI 18
  - CN6 - CONNETTORE PER INGRESSI A/D CONVERTER 20
  - CN9 - CONNETTORE INTERFACCIA ISP 22
- INTERFACCIAMENTO CONNETTORI CON IL CAMPO 26
- TRIMMER E TARATURE 26
- INTERFACCIE PER I/O DIGITALI 27
- SEGNALAZIONI VISIVE 28
- JUMPERS 28
  - JUMPERS A 2 VIE 30
  - JUMPERS A 3 VIE 30
- JUMPERS A STAGNO 31
- INTERRUPTS 31
- BACK UP 31
- SELEZIONE COMUNICAZIONE SERIALE 32
- PIN MULTIFUNZIONE 34
- RESET E WATCH DOG 36
- PROGRAMMAZIONE ISP 38

DESCRIZIONE SOFTWARE 39

MAPPAGGI ED INDIRIZZAMENTI 40
MAPPAGGIO DELLE RISORSE DI BORDO 40
MAPPAGGIO ABACO® I/O BUS 40
MAPPAGGIO MEMORIE 41
MAPPAGGIO PERIFERICHE 43

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO 44
INGRESSO DI CONFIGURAZIONE 44
DISABILITAZIONE SEGNALI DI CONTROLLO 44
REAL TIME CLOCK 45
PERIFERICHE DELLA CPU 46

SCHEDE ESTERNE 47

BIBLIOGRAFIA 50

APPENDICE A: MONTAGGIO MECCANICO A-1

APPENDICE B: DESCRIZIONE COMPONENTI DI BORDO B-1

APPENDICE C: SCHEMI ELETTRICI C-1

APPENDICE D: INDICE ANALITICO D-1

INDICE DELLE FIGURE

FIGURA 1: SCHEMA A BLOCCHI 5
FIGURA 2: FOTO SCHEDA 9
FIGURA 3: CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP 10
FIGURA 4: CN1 - CONNETTORE PER ABACO® I/O BUS 11
FIGURA 5: CN3 - CONNETTORE PER LINEA SERIALE 12
FIGURA 6: SCHEMA DI COMUNICAZIONE SERIALE 13
FIGURA 7: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232 14
FIGURA 8: ESEMIPO COLLEGAMENTO PUNTO PUNTO IN RS 422 14
FIGURA 9: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485 14
FIGURA 10: ESEMPIO COLLEGAMENTO IN RETE IN RS 485 15
FIGURA 11: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI 16
FIGURA 12: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI 16
FIGURA 13: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP 17
FIGURA 14: CN5 - CONNETTORE PER I/O DIGITALI 18
FIGURA 15: SCHEMA DI COLLEGAMENTO I/O DIGITALI 19
FIGURA 16: CN6 - CONNETTORE PER INGRESSI A/D CONVERTER 20
FIGURA 17: SCHEMA DI COLLEGAMENTO A/D CONVERTER 21
FIGURA 18: CN9 - CONNETTORE INTERFACCIA ISP 22
FIGURA 19: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE MP AVR-51+ 23
FIGURA 20: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE SEP 40 23
FIGURA 21: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE UISP-S4 24
FIGURA 22: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE STK 300 24
FIGURA 23: DISPOSIZIONE CONNETTORI, TRIMMER, BATTERIA, MEMORIE, ECC. 25
FIGURA 24: TABELLA DELLE SEGNALAZIONI VISIVE 28
FIGURA 25: TABELLA RIASSUNTIVA JUMPERS 28
FIGURA 26: DISPOSIZIONE JUMPERS 29
FIGURA 27: TABELLA JUMPERS A 2 VIE 30
FIGURA 28: TABELLA JUMPERS A 3 VIE 30
FIGURA 29: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE 33
FIGURA 30: TABELLA PIN MULTIFUNZIONE 35
FIGURA 31: PIANTA COMPONENTI LATO COMPONENTI 37
FIGURA 32: PIANTA COMPONENTI LATO STAGNATURE 37
FIGURA 33: MAPPAGGIO DELLE MEMORIE 41
FIGURA 34: ACCESSO MEMORIE 42
FIGURA 35: TABELLA INDIRIZZAMENTO PERIFERICHE ESTERNE 43
FIGURA 36: SCHEMA DELLE POSSIBILI CONNESSIONI 49
FIGURA A1: DIMA DI FORATURA PER MONTAGGIO IN PIGGY BACK A-1
FIGURA A2: MONTAGGIO IN PIGGY BACK A-2
FIGURA A3: MONTAGGIO SU GUIDA WEIDMULLER A-2
FIGURA C1: SCHEMA ELETTRICO DI ESPANSIONE PPI C-1
FIGURA C2: SCHEMA ELETTRICO SPA 03 ..... C-2
FIGURA C3: SCHEMA ELETTRICO QTP 16P C-3
FIGURA C4: SCHEMA ELETTRICO QTP 24P 1/2 ..... C-4
FIGURA C5: SCHEMA ELETTRICO QTP 24P 2/2 ..... C-5
FIGURA C6: SCHEMA ELETTRICO IAC 01 ..... C-6

FIGURA C7: SCHEMA ELETTRICO KDX X24 ..... C-7
FIGURA C8: SCHEMA ELETTRICO INTERFACCIA BUS C-8

INTRODUZIONE

L'uso di questi dispositivi é rivolto - IN VIA ESCLUSIVA - a personale specializzato.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - IN VIA ESCLUSIVA - ad un utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l'ambiente, interpretando un'elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall'ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tali dispositivi, é conveniente che l'utente - PRIMA DI COMINCIARE AD OPERARE - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

VERSIONE SCHEDA

Il presente manuale è riferito alla scheda **GPC® AM4** versione **220399** e successive. La validità delle informazioni riportate è quindi subordinata al numero di versione della scheda in uso e l'utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Sulla scheda il numero di versione è riportato in più punti sia a livello di serigrafia che di stampato (ad esempio tra la CPU ed il quarzo sia sul lato componenti che sul lato stagnature).

INFORMAZIONI GENERALI

La scheda **GPC® AM4**, che fa parte della **Serie 4** delle CPUcon ingombro di 100x50 mm, é un potente e veloce modulo di controllo, della fascia a **basso prezzo**, in grado di funzionare autonomamente come periferica intelligente e/o remotata in una più vasta rete di telecontrollo e/o di acquisizione.
La **GPC® AM4** può essere fornita di un supporto in plastica provvisto degli attacchi per le guide ad **Omega** tipo **DIN 46277-1** e **DIN 46277-3**. In questo modo non é necessario l'uso di un rack, ma la scheda può essere montata, in modo più economico, direttamente nel quadro elettrico. Viste le ridotte dimensioni della scheda, la si può montare nella stessa guida in plastica che contiene le periferiche di I/O formando in questo modo un unico elemento **BLOCK**. Un'altra tipica applicazione della scheda **GPC® AM4**, é quella di essere adoperata come un modulo di CPU da montare in **Piggy Back** sulle schede periferiche realizzate direttamente dall'utente.
L'aspetto più interessante di questa scheda é il µP utilizzato: l'**ATMEL ATmega 103** é infatti un velocissimo microcontrollore RISC che integra una ricca serie di risorse adatte a risolvere i classici problemi dell'automazione industriale.
Attualmente sono disponibili diversi tools di sviluppo software che consentono di poter usare la scheda sia in assembler che con linguaggi evoluti. Una particolare menzione và ai vari compilatori **C** e **BASIC**, che eseguiti su un normale P.C. consentono di generare codice eseguibile per la scheda. Questo codice viene comodamente scaricato nella **FLASH EPROM** interna del **µP** tramite comodi programmatori **ISP**, che a sua volta sono collegati in seriale (COM) o parallelo (LPT), al PC di sviluppo.
Per una rapida prototipizzazione si può ricorrere alle ottime schede **SPA 03** ed **SPA 04** su cui é possibile montare, in Piggy Back, la **GPC® AM4**. La presenza del connettore **ABACO® I/O BUS** consente inoltre di poter pilotare direttamente le schede periferiche tipo **ZBR xxx**, **ZBT xxx**, **CAN 14**, **ADC 812**, **DAC 212**, ecc. e tramite **l'ABB 03**, **ABB 05** é possibile gestire tutte le numerose schede periferiche disponibili sul **BUS ABACO®**.

- Modulo Intelligente **Abaco® BLOCK**, della **serie 4**, nel formato 100x50 mm.
- Contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**.
- CPU **ATmega103** da **5,5296 MHz** con:
  - **128K FLASH EPROM** interna;
  - **4K RAM** interna;
  - **4K EEPROM** interna;
  - **8** linee di **A/D converter** da 10 bit, 0÷2.49 V, su connettore standard;
  - **2 timer counter** ad 8 bit in grado di generare seganli **PWM**;
  - **1 timer counter** a 16 bit in grado di generare due segnali **PWM** da 10 bit;
  - **Watch Dog** interamente gestibile via software;
  - **Comparatore** analogico;
  - **15** sorgenti di **Interrupt** (interne ed esterne).
- **32K SRAM** esterna su zoccolo.
- **Real Time Clock** in grado di gestire giorno, mese, anno, giorno della settimana, ore, minuti, secondi e di emettere interrupt con cadenze definibili da software.
- Circuiteria di **back up** per SRAM esterna ed RTC, con batteria al **LITIO**.
- Fino a **20** linee di **I/O** TTL su connettori standard.
- **Jumpers** di configurazione acquisibile via software.
- 1 linea seriale in **RS232**, settabile in **RS422**, **RS485** o **current loop**.
- Interfaccia per **ABACO® I/O BUS** su connettore standard da 26 vie.
- Interfaccia **ISP** compatibile con programmatori **grifo® MP AVR-51+** e **SEP 40** o **UISP-S4** della EQUINOX.

- Possibilità di funzionamento in modalità **Idle** o **Power Down**.
- Unica alimentazione esterna da **5 Vdc**, **50 mA**, protetta tramite **TransZorb™**.
- Vasta disponibilità di software di sviluppo quali **Assembler**, **AVR Studio**, **DDS MICRO-C AVR**; **ICC AVR**; **BASCOM AVR**; ecc.

Viene di seguito riportata una descrizione dei blocchi funzionali della scheda, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alla figura 1.

ALIMENTAZIONE

La scheda deve essere alimentata solo con una tensione di **+5 Vdc da applicare tra i pin 25 (GND) e 26 (+5Vdc) del connettore CN1**. Il circuito di alimentazione della scheda genera tutte le tensioni necessarie ed é progettata per ridurre il consumo (sono disponibili anche le modalità a basso consumo del microprocessore) e per aumentare l'immunità ai disturbi elettrici. Infatti solamente 48 mA sono necessari per il normale funzionamento e consentono all'utente di alimentare la scheda con batterie, pannelli solari, piccoli alimentatori, ecc. Si ricorda inoltre che sulla scheda é presente una circuiteria che protegge dai picchi e dalle inversioni di alimentazione, basata su **TransZorb™**.

COMUNICAZIONE SERIALE

La comunicazione seriale é completamente settabile via software per quanto riguarda il protocollo fisico con un baud rate da 2400 a 115200, 8 o 9 bit per carattere, e vari controlli di ricezione. Tali settaggi avvengono tramite la programmazione dei registri interni del microprocessore relativi alla sezione UART come descritto nella documentazione tecnica della casa costruttrice o nell'appendice B di questo manuale.
Dal punto di vista hardware, tramite una serie di comodi jumpers e driver da installare, é invece possibile selezionare il protocollo elettrico di comunicazione. In particolare si può bufferare la linea in **RS 232**, **current loop** o **RS 422**, **RS 485**; in quest'ultimi casi é definibile anche l'attivazione e/o la direzionalità della linea di comunicazione. Si ricorda che la scheda viene normalmente fornita con la linea seriale bufferata in RS 232 e che tutte le rimanenti configurazioni devono essere quindi opportunamente specificate in fase di ordine della scheda.
Per ulteriori informazioni in merito alla comunicazione seriale fare riferimento ai paragrafi "CONNESSIONI" e "SELEZIONE COMUNICAZIONE SERIALE".

CLOCK

Sulla **GPC® AM4** sono presenti due circuiterie di clock:
- La prima é basata su un quarzo che provvede a generare la frequenza di clock per la CPU da cui vengono ricavate anche le frequenze necessarie per le altre sezioni della scheda (Timer, Seriale, SPI ecc.). Il valore standard di clock é di **5,5296 MHz**, ma in caso di applicazioni particolari, tale frequenza può essere variata: contattare direttamente **grifo®**. Si ricorda che la frequenza di clock della CPU coincide con quella del quarzo montato a bordo scheda e che si può intervenire via software sull'apposita sezione di power management per ridurne i valori e quindi i consumi.
- La seconda circuiteria provvede a fornire le giuste temporizzazioni al real time clock di bordo ed é basata su un quarzo da **32,768 KHz**, integrato nello stesso componente.

PROCESSORE DI BORDO

La scheda **GPC® AM4** é predisposta per accettare il processore **AVR Mega 103** prodotto dalla **ATMEL**. Tale processore ad 8 bit é codice compatibile con la famiglia RISC AVR ed é quindi caratterizzato da: un buon set di istruzioni (circa 120 che includono istruzioni aritmetiche, logiche, di controllo e salto, trasferimento dati e di manipolazione diretta del singolo bit), un'altissima velocità di esecuzione, un'efficiente gestione degli interrupts vettorizzati, una vasta scelta di modalità d'indirizzamento. Di fondamentale importanza é la presenza delle seguenti periferiche interne al microprocessore:

- 32 x 8 registri di lavoro a scopo generale
- 128K bytes di FLASH EPROM programmabile in system
- 4K bytes di SRAM
- 4K bytes di EEPROM programmabile in system
- Comparatore analogico
- Timer programmabile con funzionalità di watch dog
- Linea serial UART programmabile
- Interfaccia seriale SPI di tipo Master/Slave
- Due timer counter ad 8 bit con divisori separati e modalità PWM
- Un timer counter espanso a 16 bit con divisore separato, modalià di comparazione e cattura, doppio PWM da 8/9/10 bits
- 8 canali di A/D converter da 10 bit
- 32 linee di I/O programmabili
- Tre diverse modalità a basso consumo
- Clock di lavoro selezionabile via software
- Numerose sorgenti d'interrupt sia interne che esterne

Per maggiori informazioni sul componente si faccia riferimento all'apposita documentazione della casa costruttrice, oppure all'appendice B di questo manuale.

ABACO® I/O BUS

Una delle caratteristiche di fondamentale importanza della **GPC® AM4** è quella di disporre del cosiddetto **ABACO®** I/O BUS: ovvero un connettore normalizzato **ABACO®** con cui è possibile collegare la scheda ad una serie di moduli esterni intelligenti e non. Tra questi si trovano moduli per acquisizione di segnali analogici (A/D), per la generazione di segnali analogici (D/A), schede con contatori e temporizzatori, schede per gestione di linee di I/O logico, ecc. e ne possono essere realizzati anche su specifiche richieste dell'utente.
Utilizzando mother board come l'**ABB 03** o l'**ABB 05** é inoltre possibile gestire tutte le schede periferiche in formato Europa con interfaccia per BUS **ABACO®**. Tale caratteristica rende la scheda espandibile con un ottimo rapporto prezzo/prestazioni e quindi adatta a risolvere molti dei problemi dell'automazione industriale.
Da ricordare che le linee dell'**ABACO®** I/O BUS, conformemente alle caratteristiche della serie 4, non sono bufferate e quindi l'utente deve provvedere ad una adeguata protezione contro i disturbi esterni ed il comando di un limitato numero di schede periferiche di espansione.
Il capitolo "SCHEDE ESTERNE" mostra una breve descrizione di alcune di queste schede di espansione.

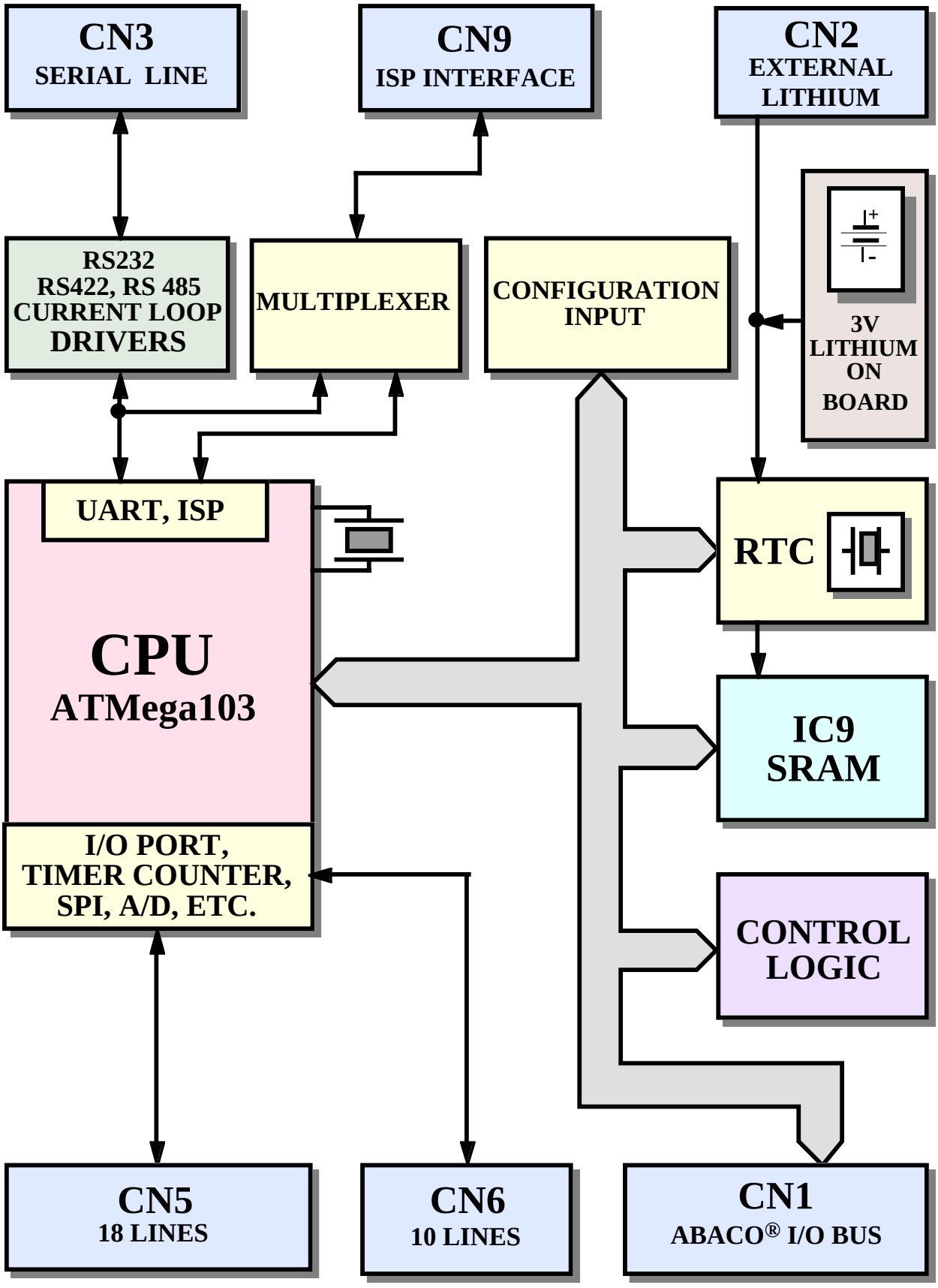


FIGURA 1: SCHEMA A BLOCCHI

MEMORIE

Sulla scheda sono disponibili 164K Bytes di memoria suddivisi tra 128K Byte di FLASH EPROM, 32K SRAM ed infine 4K Byte di EEPROM. La scelta dell'uso delle memorie presenti sulla scheda può avvenire in relazione all'applicazione da risolvere e quindi alle esigenze dell'utente. Da questo punto di vista si ricorda che la scheda viene normalmente fornita con tutta la memoria descritta e che questa non può essere incrementata e nemmeno ridotta.
Sfruttando la circuiteria di back up di bordo più l'eventuale batteria tampone esterna, si ha la possibilità di mantenere i dati anche in assenza di alimentazione. Questa caratteristica fornisce alla scheda la possibilità di ricordare in ogni condizione, una serie di parametri come ad esempio la configurazione o lo stato del sistema, anche per lunghi periodi di inattività senza dover ricorrere a costosi gruppi di continuità esterni.
Il mappaggio delle risorse di memoria é direttamente controllata dal microprocessore che provvede ad allocare i dispositivi all'interno del suo spazio d'indirizzamento. Questa caratteristica consente di gestire automaticamente tutte le modalità d'indirizzamento richieste dai pacchetti software per la **GPC® AM4**, senza richiedere un intervento diretto dell'utente.
Per maggiori informazioni fare riferimento al capitolo "MAPPAGGI ED INDIRIZZAMENTI".

REAL TIME CLOCK

La **GPC® AM4** dispone di un completo Real Time Clock in grado di gestire ore, minuti, secondi, giorno del mese, mese, anno e giorno della settimana in modo completamente autonomo. Il componente é alimentato dalla circuiteria di back up in modo da garantire la validità dei dati in ogni condizione operativa ed é completamente gestito via software, tramite la programmazione di 16 registri situati nell'area dati della CPU dall'apposita logica di controllo. La sezione di RTC può inoltre generare interrupt in corrispondenza di intervalli di tempo programmabili via software, in modo da poter periodicamente distogliere la CPU dalle normali operazioni oppure periodicamente risvegliarla dagli stati di basso consumo.

WATCH DOG

La scheda **GPC® AM4** é provvista di una circuiteria di watch dog che, se utilizzata, consente di uscire da stati di loop infinito o da condizioni anomale non previste dal programma applicativo. Tale circuiteria é composta da una sezione astabile con un tempo d'intervento programmabile da 12 a 1600 mse, é completamente gestibile via software (tramite l'accesso ad un apposito registro interno della CPU) e conferisce al sistema basato sulla scheda, una sicurezza estrema in qualsiasi condizione operativa.

A/D CONVERTER

La sezione analogica della **GPC® AM4** é basata su un convertitore A/D interno da 8 canali con 10 bit di risoluzione massima. Da software l'utente può selezionare il canale e la velocitàda da usare, avviare e terminare la conversione, ed abilitare l'interrupt di fine conversione, tramite 4 registri interni al microprocessore. Gli ingressi analogici possono essere nel range 0÷2,490 V, come specificato nel paragrafo "TRIMMER E TARATURE" e sono tutti dotati di una capacità di filtro che aumenta l'immunità ai disturbi e la stabilità dei dati acquisiti.

LINEE DI I/O DIGITALI

La scheda dispone di 20 linee di I/O digitale a livello TTL con direzionalità settabile per ogni linea, direttamente gestite dal microprocessore. Questi segnali sono collegati a due connettori con pin out standard **grifo®** (18 linee con standard **I/O ABACO®** e 2 con standard **A/D ABACO®**) e possono essere quindi direttamente collegati alle numerose schede provviste dello stesso pin out.
Via software é definibile la funzionalità di queste linee, con possibilità di associarle anche alle periferiche della scheda (SPI, Timer Counter, Interrupt, ecc.), tramite una semplice programmazione di 9 registri interni della CPU.
Per maggiori informazioni fare riferimento ai paragrafi "CONNESSIONI", "INTERFACCE PER I/O DIGITALI", "PIN MULTIFUNZIONE" e "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO".

TIMER COUNTER

La **GPC® AM4** é provvista di 3 timer counter ad uso generico, di cui 2 ad 8 bits ed uno a 16 bits. Per ognuno può essere definito il divisore e la modalità di funzionamento via software grazie a 18 appositi registri interni del microprocessore. I timer counter possono funzionare come temporizzatori basati su una frequenza di clock interna, come contatori di segnali esterni e come generatori di segnali modulati in frequenza (PWM). Per ulteriori informazioni vedere il paragrafo "PIN MULTIFUNZIONE" e l'appendice B di questo manuale.

INTERFACCIA ISP

La scheda supporta la tecnica di programmazione in system (ISP) che consente di leggere e/o scrivere le aree codice e dati interne al microprocessore, tramite una semplice connessione seriale e quindi senza rimuovere alcun componente dalla **GPC® AM4**. Questa programmazione seriale é basata sul protocollo standard industriale SPI che coincide con un BUS sincrono a tre file di cui due per i dati (ricezione e trasmissione) ed uno per il clock.
L'interfaccia ISP della scheda é compatibile con la maggioranza dei programmatori in commercio provvisti della stessa interfaccia, come ad esempio l'**MP AVR-51+** ed il **SEP 40** della **grifo®** (collegaili al PC di sviluppo tramite la porta parallela LPT) o l'**UISP-S4** della EQUINOX (collegaile al PC di sviluppo tramite la porta seriale COM). I paragrafi "CN9 - CONNETTORE PER INTERFACCIA ISP" e "PROGRAMMAZIONE ISP" fornisco ulteriori informazioni relative al collegamento ed all'uso di questa interfaccia.

SPECIFICHE TECNICHE

CARATTERISTICHE GENERALI

Risorse della scheda:	20 input/output digitali TTL		
	2 timer counter a 8 bit		
	1 timer counter a 16 bit		
	4 segnali PWM		
	1 linea seriale RS 232, RS 422, RS 485, current loop		
	8 linee di A/D converter		
	1 contatto di reset		
	1 watch dog astabile		
	1 real time clock		
	1 jumper di configurazione		
	1 circuiteria di back up		
	1 interfaccia **ABACO®** I/O BUS		
	1 interfaccia ISP		
Memoria indirizzabile:	IC 6:	64K x 16 (128K x 8) FLASH EPROM	
		4K x 8 static RAM	
		4K x 8 EEPROM	
	IC 9:	32K x 8 static RAM	
Tempo d'accesso memorie:	100 nsec		
CPU di bordo:	ATMEL AVR Mega 103		
Frequenza clock CPU:	5,5296 MHz		
Frequenza max timer counter:	Frequenza clock CPU		
Frequenza taglio ingressi A/D:	1 MHz		
Risoluzione A/D:	10 bit		
Tempo conversione A/D:	5÷648 µsec	(bassa precisione)	
	87÷324 µsec	(alta precisione)	
Errore complessivo A/D:	±2 punti		
Tempo intervento watch dog:	12÷1600 msec programmabile via software		

CARATTERISTICHE FISICHE

Dimensioni (L x A x P):	100 x 50 x 25 mm	(senza contenitore)	
	110 x 60 x 60 mm	(con contenitore per guide DIN)	
Peso:	60 g	(senza contenitore)	
	120 g	(con contenitore per guide DIN)	
Connettori:	CN1:	26 vie scatolino verticale M	
	CN2:	2 vie scatolino verticale M	
	CN3:	Plug a 6 vie 90 gradi F	
	CN5:	20 vie scatolino verticale M	
	CN6:	20 vie scatolino verticale M	
	CN9:	10 vie scatolino verticale M	
Range di temperatura:	da 0 a 50 gradi Centigradi		
Umidità relativa:	20% fino a 90%	(senza condensa)	

CARATTERISTICHE ELETTRICHE

Tensione di alimentazione:	5 Vdc ±5%	
Corrente assorbita sui +5 Vdc:	48 mA	configurazione base
	80 mA	configurazione massima
	35 mA	power down
Batteria di bordo di back up:	3,0 Vdc; 180 mAh	
Batteria esterna di back up:	3,6÷5 Vdc	
Corrente di back up:	10 µA	(batteria di bordo)
	12 µA	(batteria esterna da 3,6 Vdc)
Ingressi analogici:	0÷2,490 V	
Impedenza ingressi analogici:	100 MΩ	
Ingresso analogico comparatore:	0÷5,000 V	
Rete terminazione RS 422-485:	Resistenza terminazione linea=	120 Ω
	Resistenza di pull up sul positivo=	3,3 KΩ
	Resistenza di pull down sul negativo=	3,3 KΩ

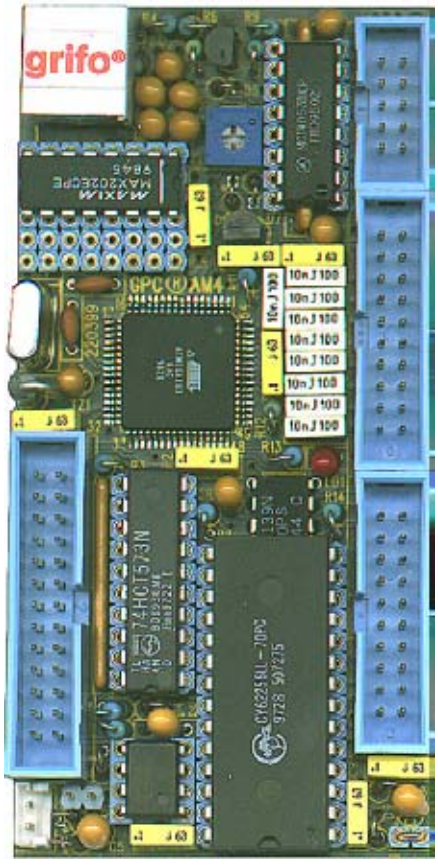


FIGURA 2: FOTO SCHEDA

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni da effettuare per il corretto utilizzo della scheda. A questo scopo viene riportata l'ubicazione e la funzione degli strip, dei connettori, dei trimmers, dei LED, ecc. presenti sulla **GPC® AM4**.

CONNESSIONI

Il modulo **GPC® AM4** è provvisto di 6 connettori con cui vengono effettuati tutti i collegamenti con il campo e con le altre schede del sistema di controllo da realizzare. Di seguito viene riportato il loro pin out ed il significato dei segnali collegati; per una facile individuazione di tali connettori, si faccia riferimento alla figura 23, mentre per ulteriori informazioni a riguardo del tipo di connessioni, fare riferimento alle figure successive che illustrano il tipo di collegamento effettuato a bordo scheda.

CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

CN2 é un connettore a scatolino, verticale, maschio, con passo 2,54 mm a 2 vie.
Tramite CN2 deve essere collegata una batteria esterna che provvede a mantenere i dati della SRAM di bordo ed a garantire il funzionamento del real time clock, in assenza di tensione di alimentazione (per maggiori informazioni fare riferimento al paragrafo "BACK UP".

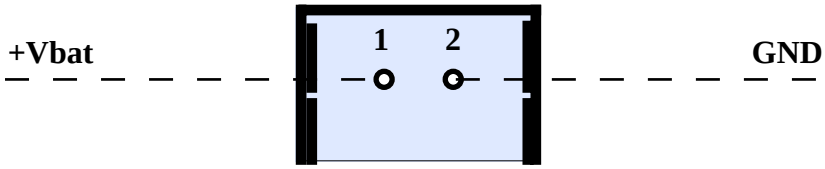


FIGURA 3: CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

Legenda:

+Vbat	=	I	- Positivo della batteria esterna di back up.
GND	=		- Negativo della batteria esterna di back up.

CN1 - CONNETTORE PER ABACO® I/O BUS

CN1 è un connettore a scatolino verticale con passo 2.54 mm a 26 piedini.
Tramite CN1 si effettua la connessione tra la scheda e la serie di moduli esterni di espansione. Tale collegamento è effettuato tramite l'**ABACO®** I/O BUS di cui questo connettore riporta tutti i segnali a livello TTL.

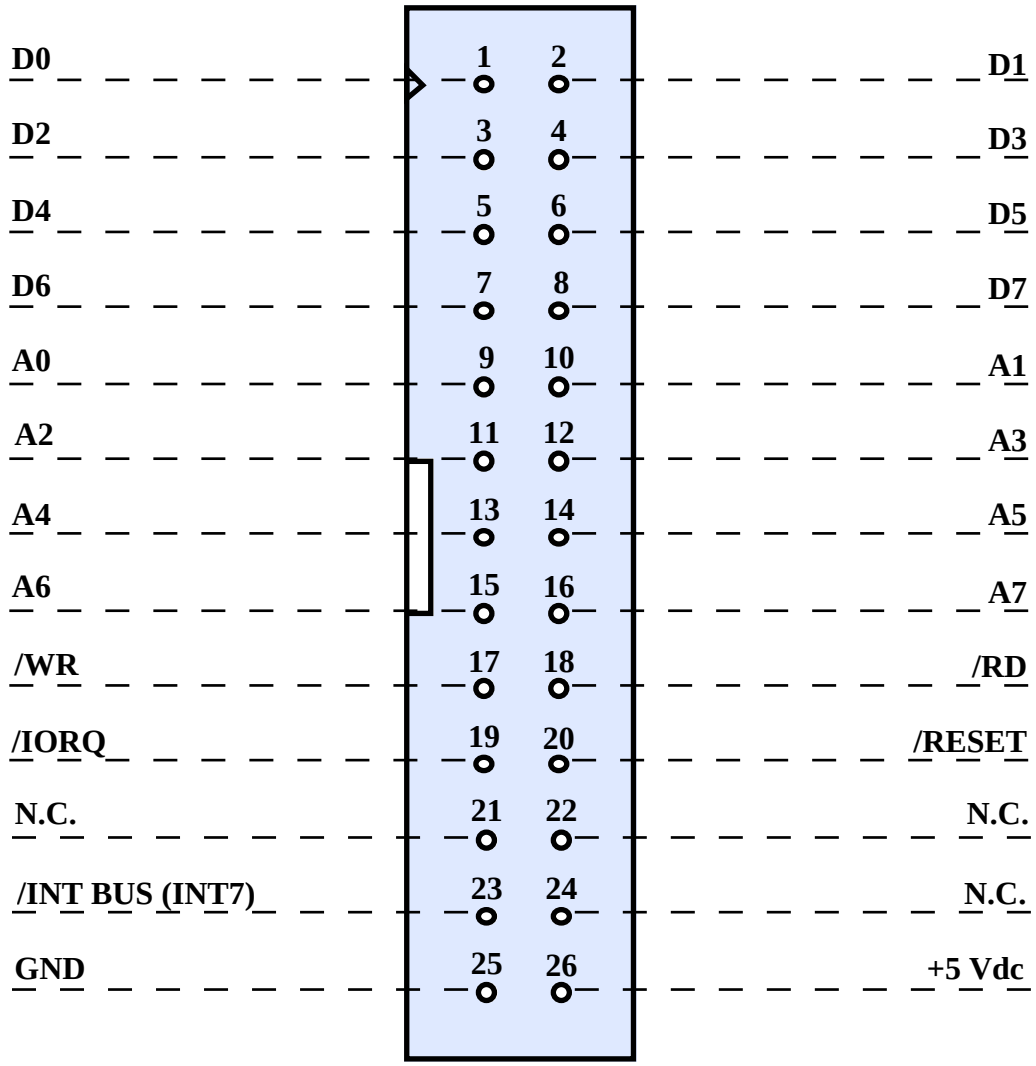


FIGURA 4: CN1 - CONNETTORE PER ABACO® I/O BUS

Legenda:

A0÷A7	= O	- Address BUS: BUS degli indirizzi.
D0÷D7	= I/O	- Data BUS: BUS dei dati.
/INT BUS	= I	- Interrupt request: richiesta d'interrupt, collegato a segnale /INT7 della CPU.
/IORQ	= O	- Input Output Request: richiesta operazione Input Output su I/O BUS.
/RD	= O	- Read cycle status: richiesta di lettura.
/WR	= O	- Write cycle status: richiesta di scrittura.
/RESET	= O	- Reset: azzeramento.
+5 Vdc	= I	- Linea di alimentazione a +5 Vdc.
GND	=	- Linea di massa.
N.C.	=	- Non collegato.

CN3 - CONNETTORE PER LINEA SERIALE

CN3 é un connettore femmina, a 90 gradi,, del tipo plug a 6 vie.
Sul connettore sono disponibili i segnali per la comunicazione della linea seriale, in RS 232, RS 422, RS 485 o current loop che é fisicamente collegata alla sezione UART della CPU. La disposizione dei segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative allo standard utilizzato.

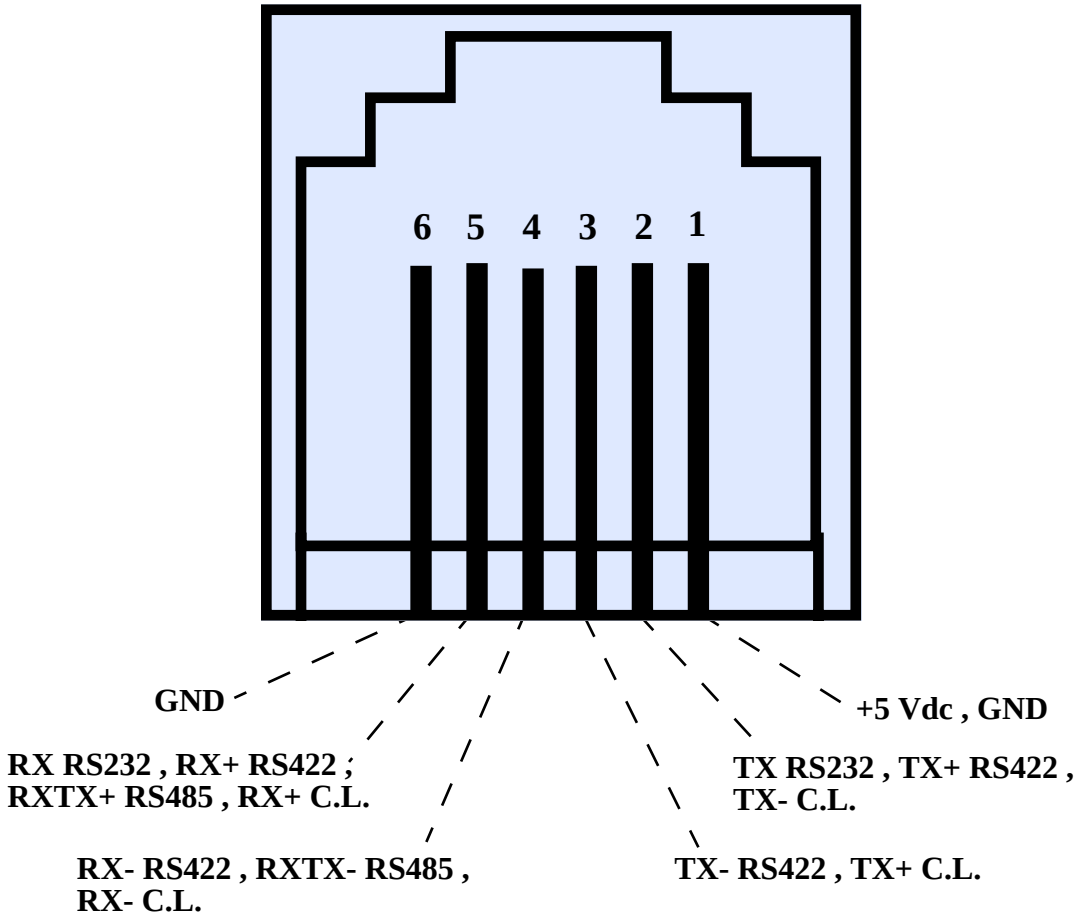


FIGURA 5: CN3 - CONNETTORE PER LINEA SERIALE

Legenda:

RX RS232	= I	- Linea ricezione in RS 232.
TX RS232	= O	- Linea trasmissione in RS 232.
RX- RS422	= I	- Linea bipolare negativa di ricezione differenziale in RS 422.
RX+ RS422	= I	- Linea bipolare positiva di ricezione differenziale in RS 422.
TX- RS422	= O	- Linea bipolare negativa di trasmissione differenziale in RS 422.
TX+ RS422	= O	- Linea bipolare positiva di trasmissione differenziale in RS 422.
RXTX- RS485	= I/O	- Linea bipolare negativa di ricezione e trasmissione differenziale in RS 485.
RXTX+ RS485	= I/O	- Linea bipolare positiva di ricezione e trasmissione differenziale in RS 485.
RX- C.L.	= I	- Linea bipolare negativa di ricezione in current loop.
RX+ C.L.	= I	- Linea bipolare positiva di ricezione in current loop.
TX- C.L.	= O	- Linea bipolare negativa di trasmissione in current loop.
TX+ C.L.	= O	- Linea bipolare positiva di trasmissione in current loop.
+5 Vdc	= O	- Linea di alimentazione a +5 Vdc.
GND	=	- Linea di massa.

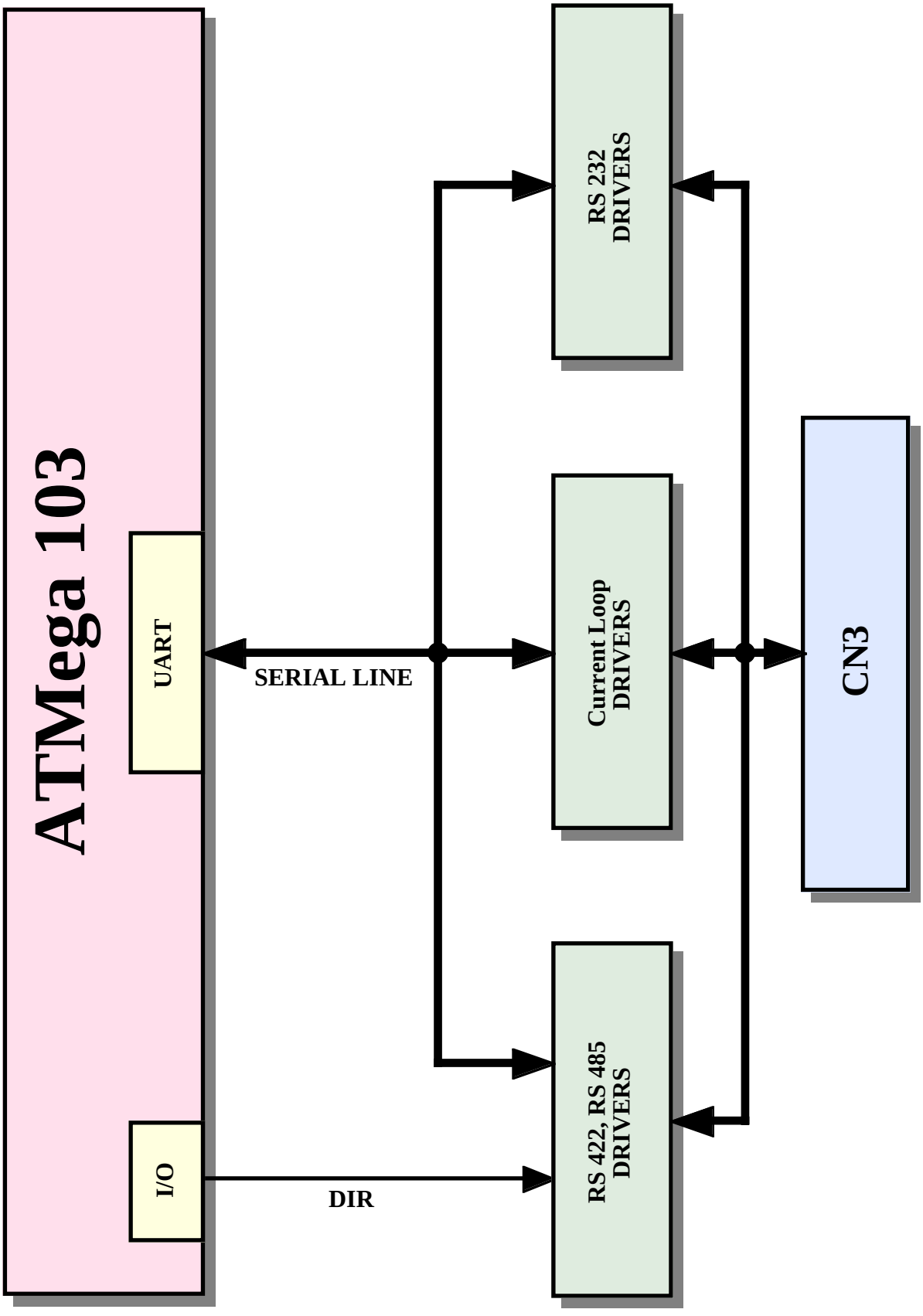


FIGURA 6: SCHEMA DI COMUNICAZIONE SERIALE

FIGURA 7: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232

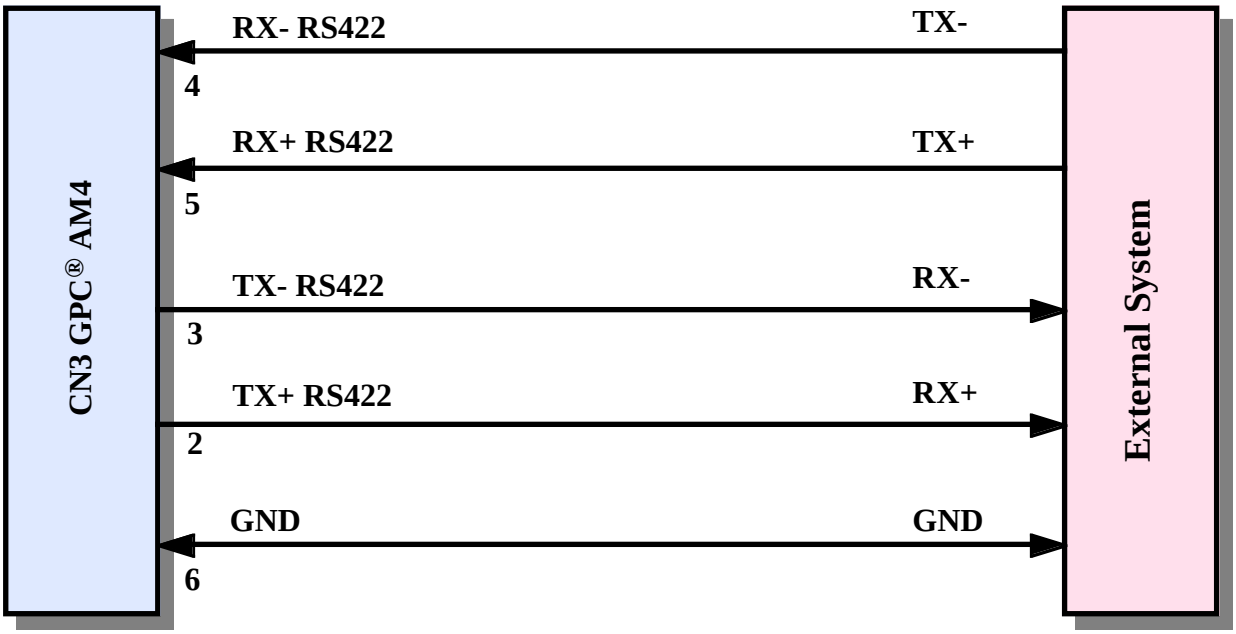


FIGURA 8: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 422

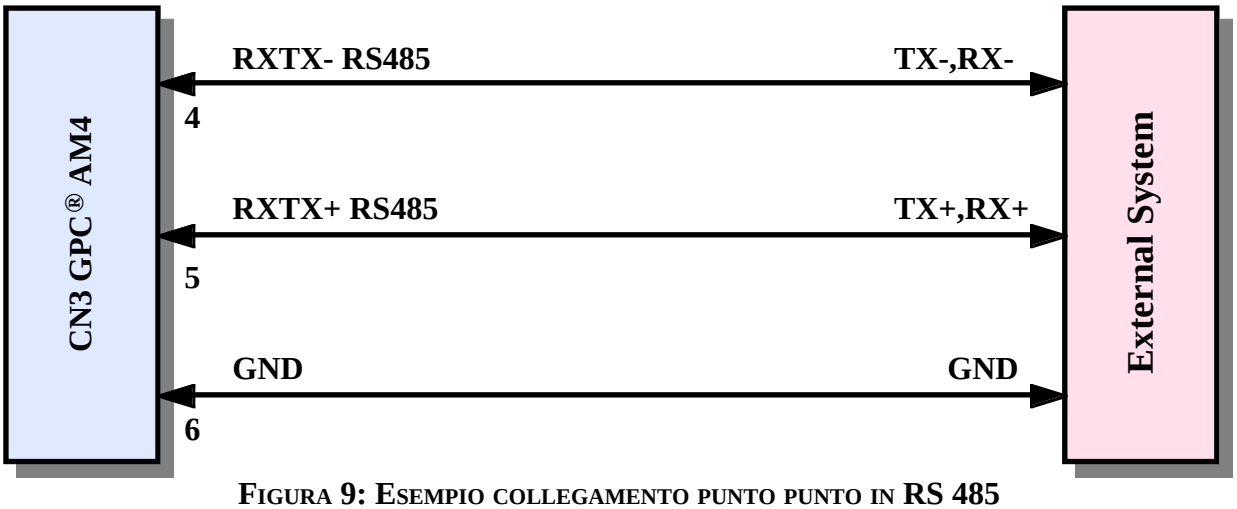


FIGURA 9: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485

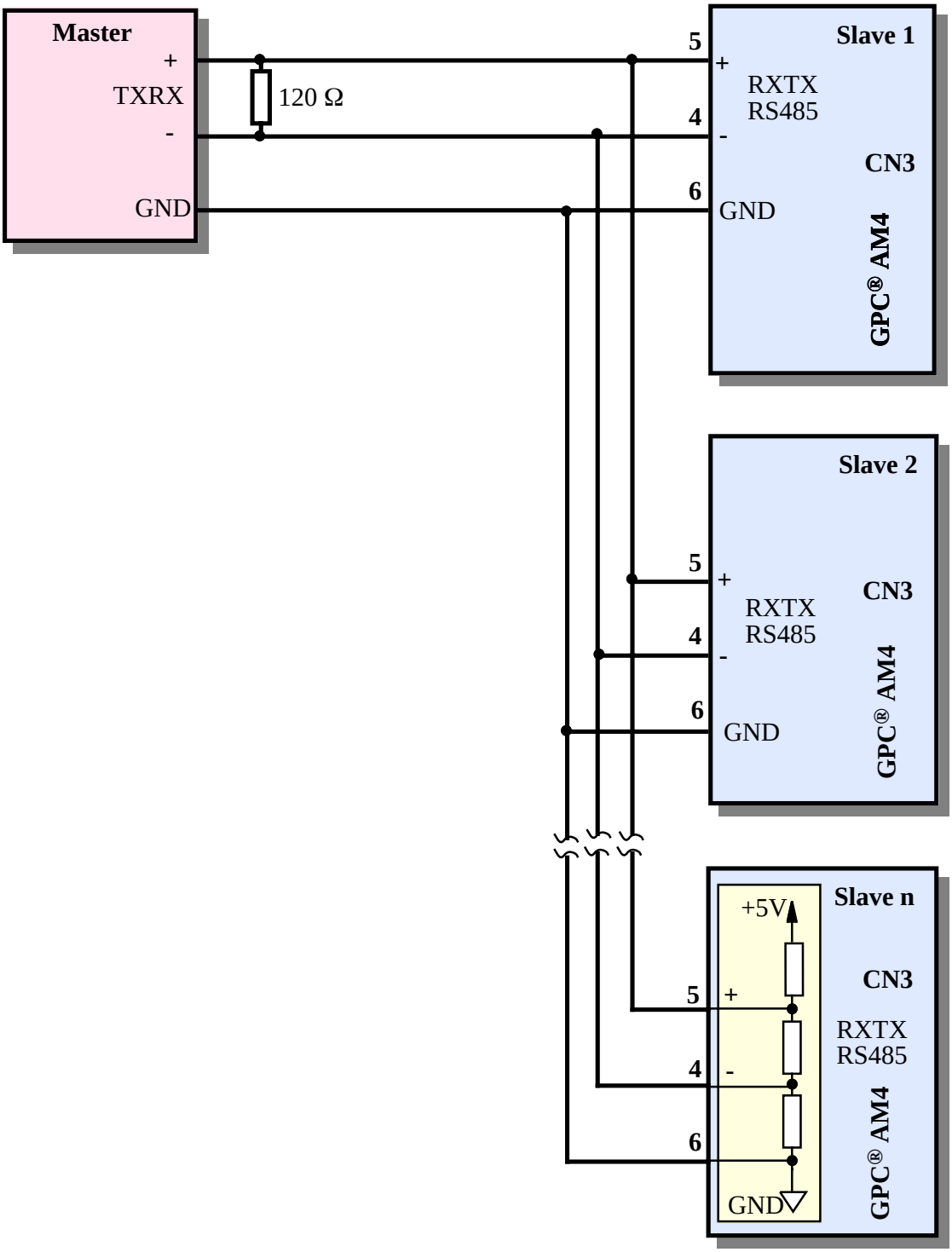


FIGURA 10: ESEMPIO COLLEGAMENTO IN RETE IN RS 485

Da notare che in una rete RS 485, devono essere presenti due resistenze di forzatura lungo la linea e due resitenze di terminazione (120 Ω), alle estremità della stessa, rispettivamente vicino all'unità Master ed all'ultima unità Slave.
A bordo della **GPC® AM4** è presente la circuiteria di terminazione e forzatura, che può essere inserita o disinserita, tramite appositi jumpers, come illutrato in seguito.
In merito alla resistenza di terminazione dell'unità Master, provvedere a collegarla solo se questa non é già presente al suo interno (ad esempio molti convertitori RS232-RS485 ne sono già provvisti).
Per maggiori informazioni consultare il Data-Book TEXAS INSTRUMENTS, "*RS 422 and RS 485 Interface Cicuits*", nella parte introduttiva riguardante le reti RS 422-485.

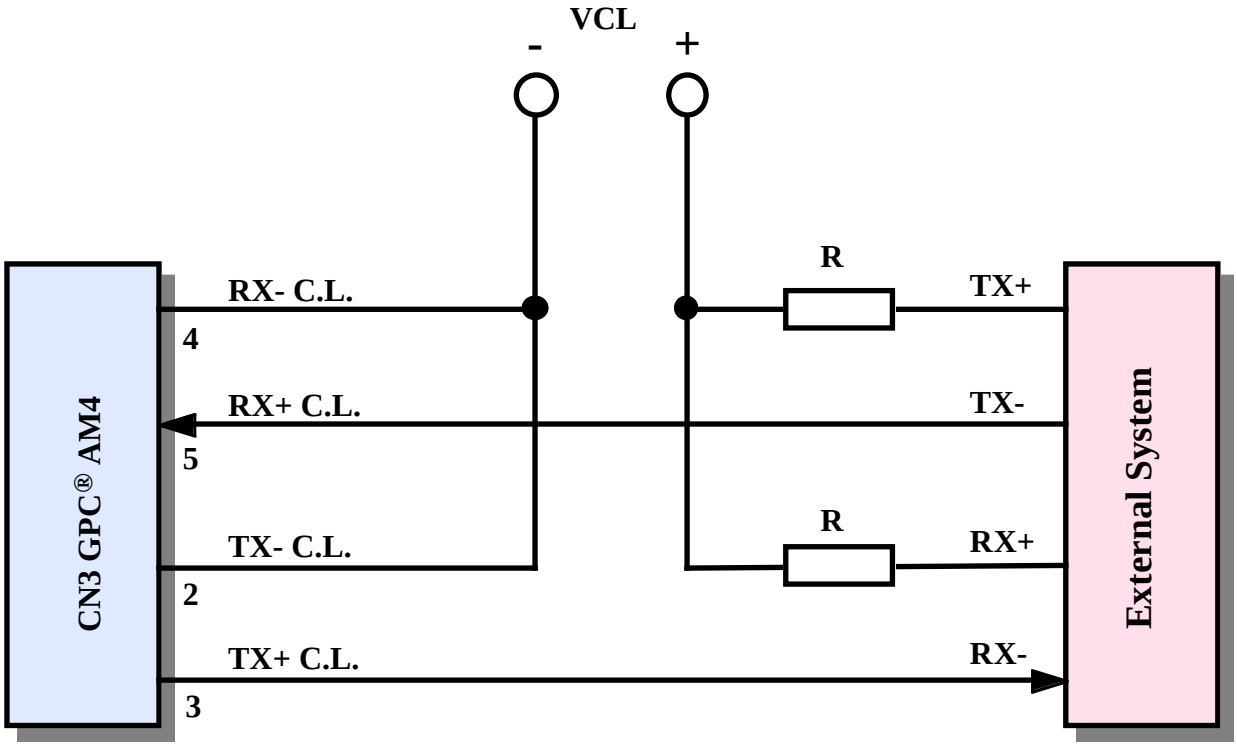


FIGURA 11: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI

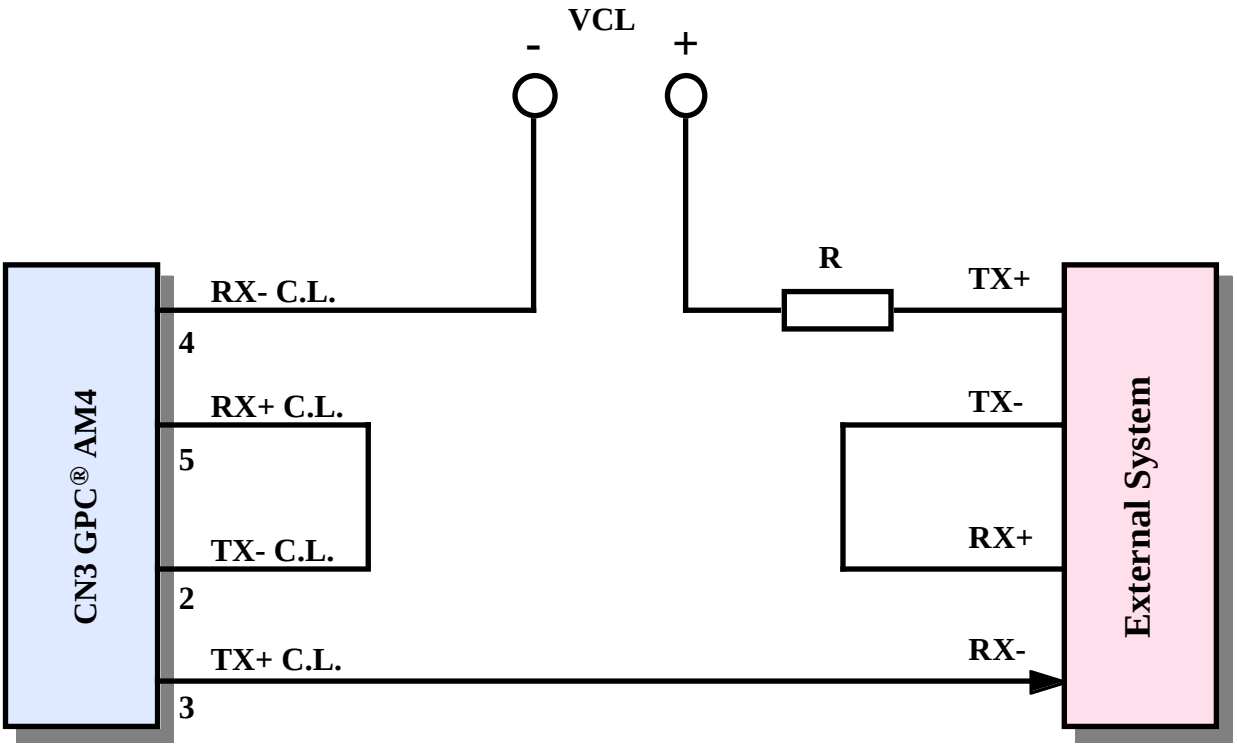


FIGURA 12: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI

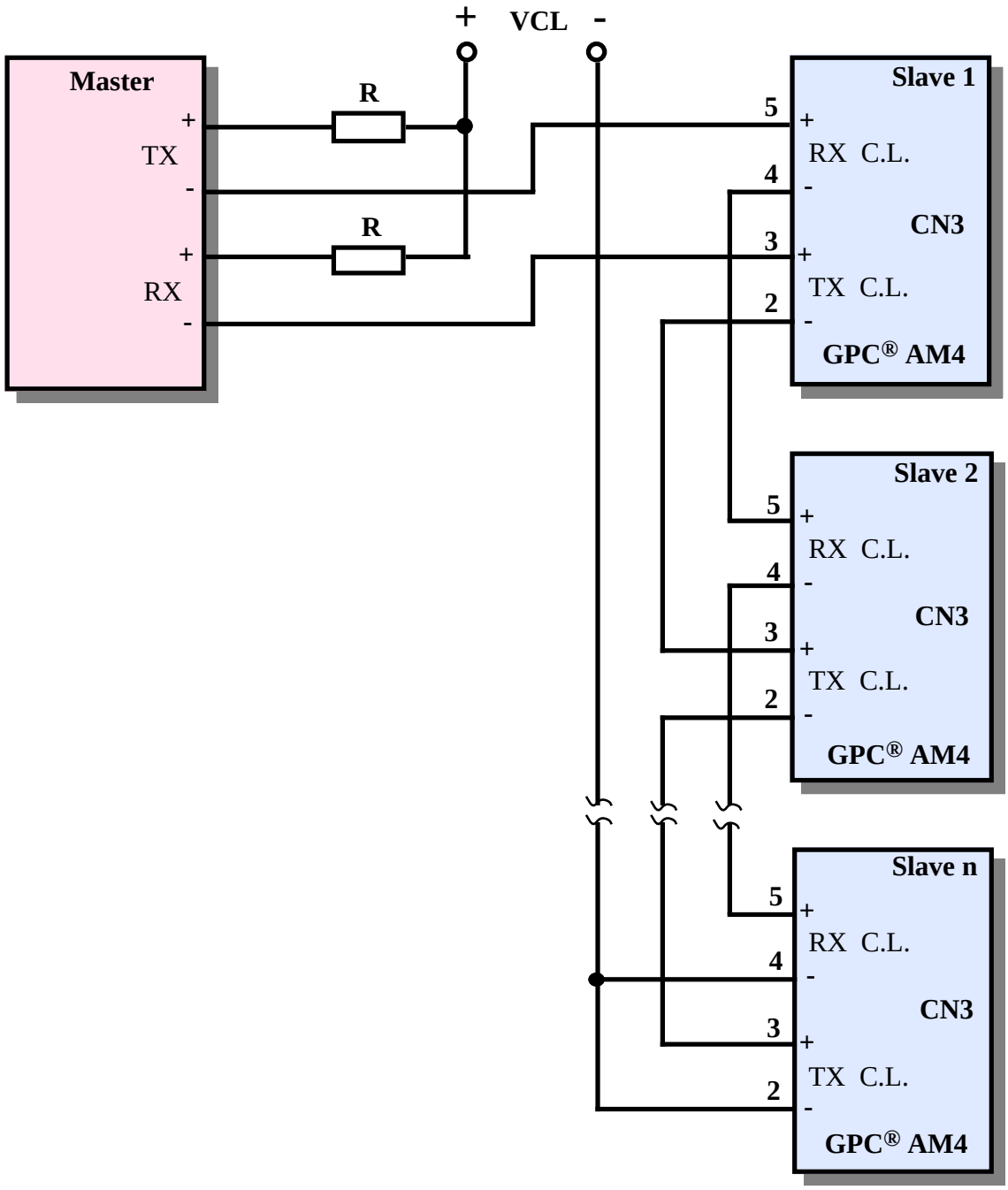


FIGURA 13: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP

Per il collegamento in current loop passivo sono possibili due diversi tipi di collegamento: a 2 fili ed a 4 fili. Tali connessioni sono riportate nelle figure 11÷13; in esse é indicata la tensione per alimentare l'anello (**VCL**) e le resistenze di limitazione della corrente (**R**). I valori di tali componenti variano in funzione del numero di dispositivi collegati e della caduta sul cavo di collegamento; bisogna quindi effettuare la scelta considerando che:
- si deve garantire la circolazione di una corrente di **20 mA**;
- su ogni trasmettitore cadono mediamente **2,35 V** con una corrente di 20 mA;
- su ogni ricevitore cadono mediamente **2,52 V** con una corrente di 20 mA;
- in caso di cortocircuito sulla rete ogni trasmettitore dissipi al massimo **125 mW**;
- in caso di cortocircuito sulla rete ogni ricevitore dissipi al massimo **90 mW**.

Per maggiori informazioni consultare il Data-Book HEWLETT-PACKARD, nella parte che riguarda gli opto accoppiatori per current loop denominati **HCPL 4100** e **HCPL 4200**.

CN5 - CONNETTORE PER I/O DIGITALI

CN5 è un connettore a scatolino verticale, maschio, con passo 2.54 mm, a 20 piedini.
Tramite CN5 si effettua la connessione tra 18 linee di I/O digitale della scheda e l'ambiente esterno. Alcuni piedini del connettore hanno una duplice funzione infatti, via software, le sezioni interne della CPU possono essere multiplexate con i segnali di I/O. Tutti i segnali di CN5 seguono il pin out standardizzato I/O **ABACO®** e coincidono con segnali logici a livello TTL eccetto i segnali del comparatore analogico AC- ed AC+ che possono essere segnali analogici variabili nel range 0÷5 V.

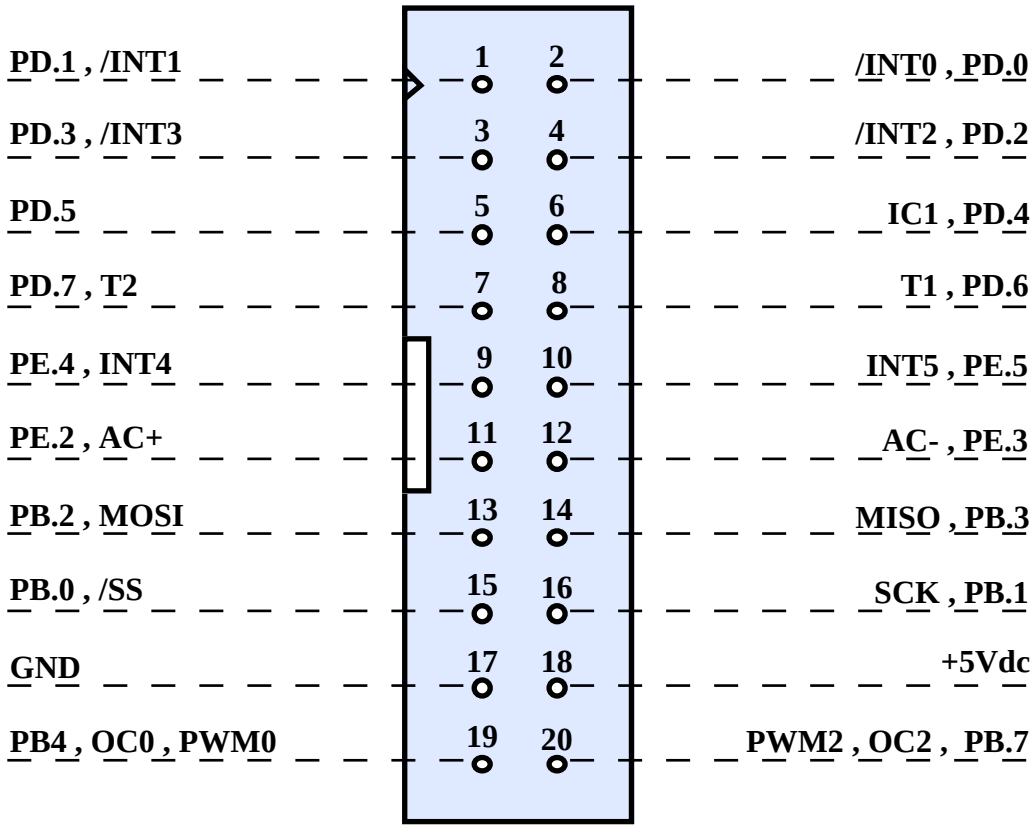


FIGURA 14: CN5 - CONNETTORE PER I/O DIGITALI

Legenda:

PB.n	= I/O	- Linea digitale n del port B della CPU.
PD.n	= I/O	- Linea digitale n del port D della CPU.
PE.n	= I/O	- Linea digitale n del port E della CPU.
/INTn , INTn	= I	- Linea di richiesta interrupt n.
Tn	= I	- Segnale d'ingresso per timer counter n.
IC1	= I	- Segnale d'ingresso per capture del timer 1.
OCn	= O	- Segnale d'uscita del timer counter n.
MOSI,MISO,/SS,SCK	= I/O	- Segnali dell'interfaccia seriale SPI.
AC+, AC-	= I	- Ingresso positivo e negativo del comparatore analogico.
PWMn	= O	- Segnale a modulazione di frequenza n.
+5 Vdc	= O	- Linea di alimentazione a +5 Vdc.
GND	=	- Linea di massa.

Per ulteriori informazioni relative ai segnali descritti si faccia riferimento al paragrafo "PIN MULTIFUNZIONE" ed all'appendice B.

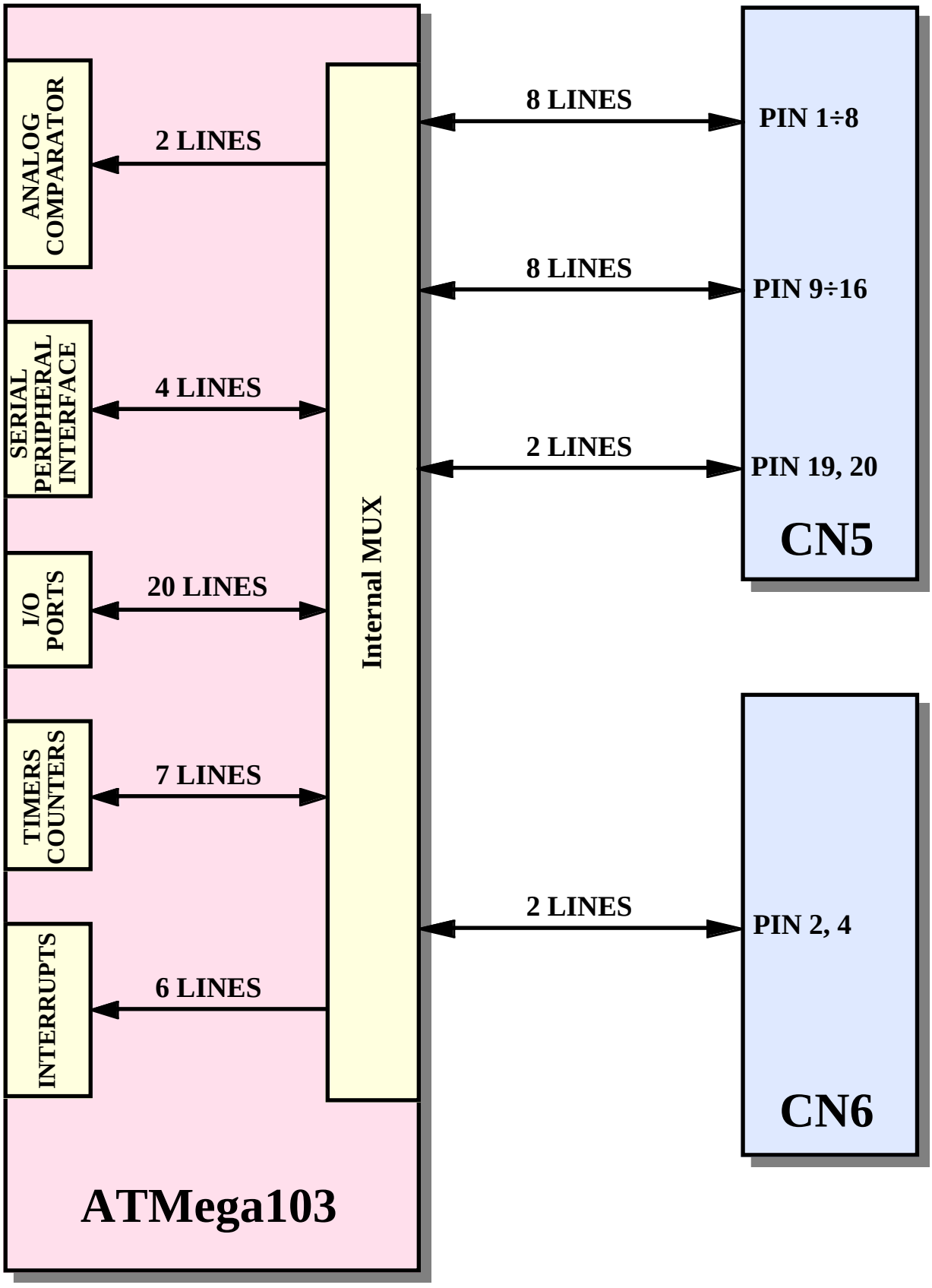


FIGURA 15: SCHEMA DI COLLEGAMENTO I/O DIGITALI

CN6 - CONNETTORE PER INGRESSI A/D CONVERTER

CN6 é un connettore a scatolino, verticale, con passo 2.54 mm a 20 piedini.
Tramite CN6 possono essere collegate le 8 linee analogiche d'ingresso con il campo esterno. Tali linee sono direttamente collegate alla sezione A/D della scheda, sono ad alta impedenza, sono provviste di un condensatore di filtro e possono variare nel range 0÷2,490 V. Su CN6 sono inoltre collegate due linee di I/O TTL multifunzione. La disposizione dei segnali su questo connettore é studiata in modo da ridurre tutti i problemi di rumore ed interferenza, garantendo quindi un'ottima trasmissione del segnale.

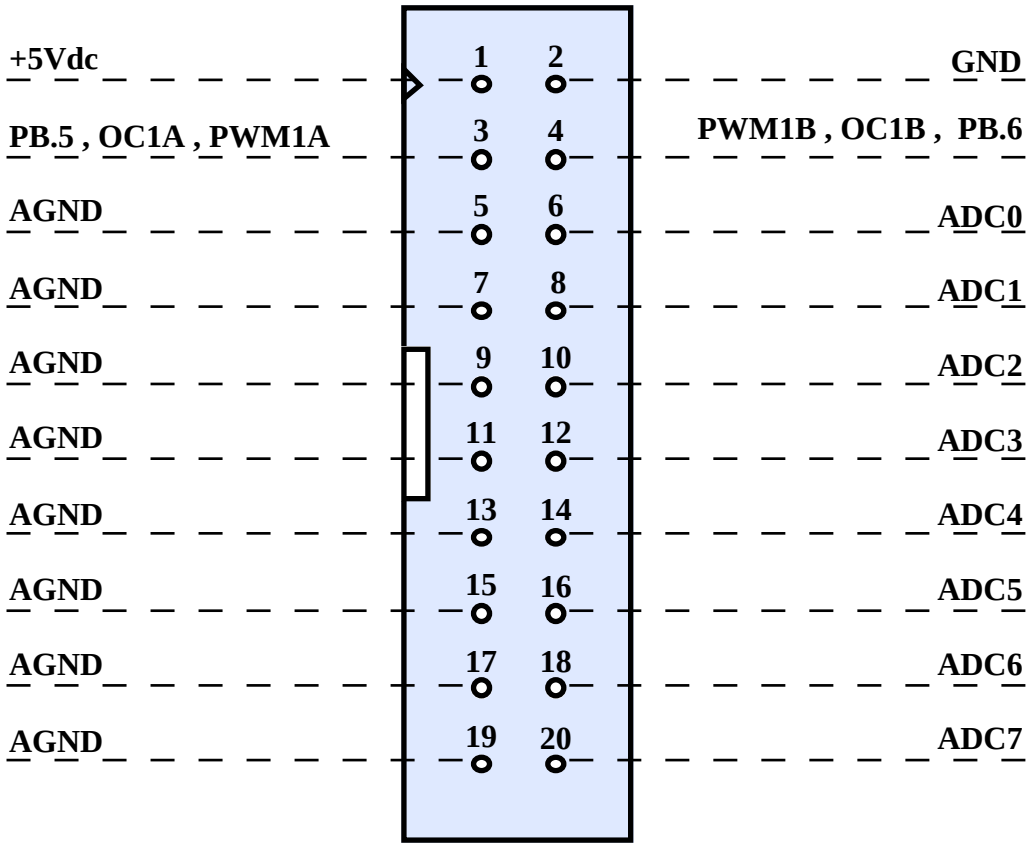


FIGURA 16: CN6 - CONNETTORE PER INGRESSI A/D CONVERTER

Legenda:

ADCn	= I	- Linea analogica d'ingresso collegata al canale n dell'A/D converter
AGND	=	- Linea di massa analogica
PB.n	= I/O	- Linea digitale n del port B della CPU.
OCnm	= O	- Segnale d'uscita del timer counter n, della sezione m.
PWMnm	= O	- Segnale a modulazione di frequenza n, della sezione m.
+5 Vdc	= O	- Linea di alimentazione a +5 Vdc
GND	=	- Linea di massa

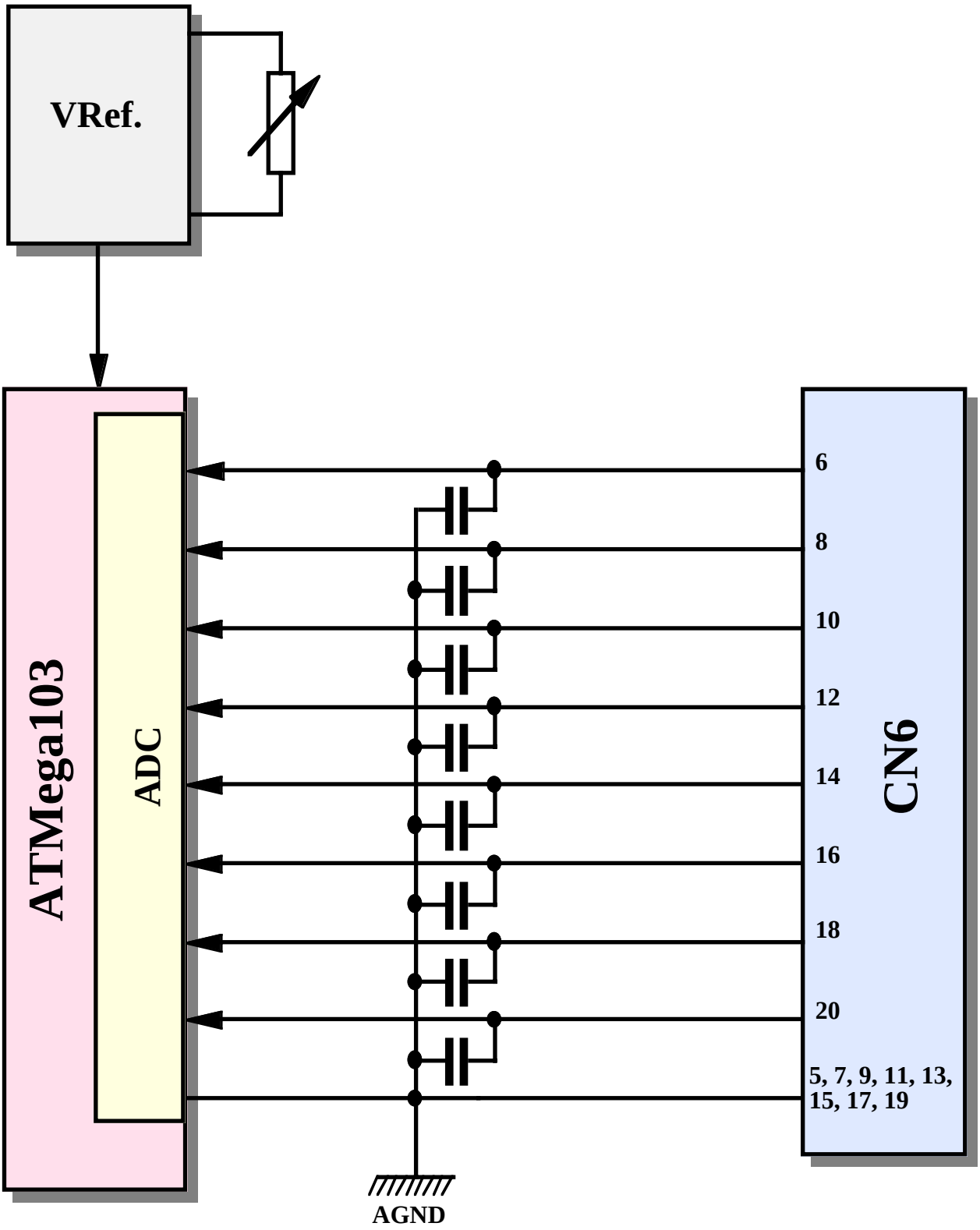


FIGURA 17: SCHEMA DI COLLEGAMENTO A/D CONVERTER

CN9 - CONNETTORE INTERFACCIA ISP

CN10 é un connettore a scatolino, verticale, con passo 2.54 mm a 10 piedini.
Tramite CN9 si può collegare l'interfaccia per la programmazione in system della scheda ai programmatori ISP esterni. Tutti i segnali di CN9 coincidono con segnali logici a livello TTL e seguono il pin out standardizzato EQUINOX.

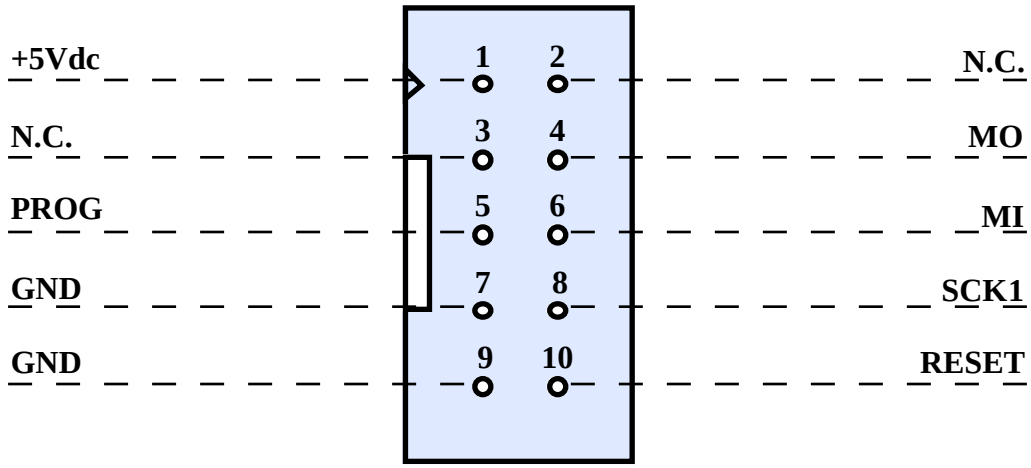


FIGURA 18: CN9 - CONNETTORE INTERFACCIA ISP

Legenda:

PROG	=	I	- Segnale selezione programmazione ISP.
MO	=	I	- Segnale master in uscita per la comunicazione ISP.
MI	=	O	- Segnale master in ingresso per la comunicazione ISP.
SCK1	=	I	- Segnale di clock seriale per la comunicazione ISP.
RESET	=	I	- Segnale di controllo RESET della scheda.
+5 Vdc	=	I	- Linea di alimentazione a +5 Vdc.
GND	=		- Linea di massa.
N.C.	=		- Non collegato.

N.B.
Per i due segnali di comunicazione MO ed MI la direzionalità indicata é opposta a quella descritta in quanto la descrizione verbale é riferita al programmatore esterno da collegare mentre la direzionalità é riferita alla scheda.
Al fine di semplificare il collegamento dei vari programmatori ISP di seguito vengono riportati alcuni esempi di connessione con i più diffusi modelli in commercio; normalmente é sufficiente smistare i segnali tra i due connettori (scheda e programmatore) senza dover aggiungere alcun componente. Per sicurezza é comunque preferibile fare riferimento anche all'apposita documentazione d'uso degli stessi programmatori ed al paragrafo "PROGRAMMAZIONE ISP".

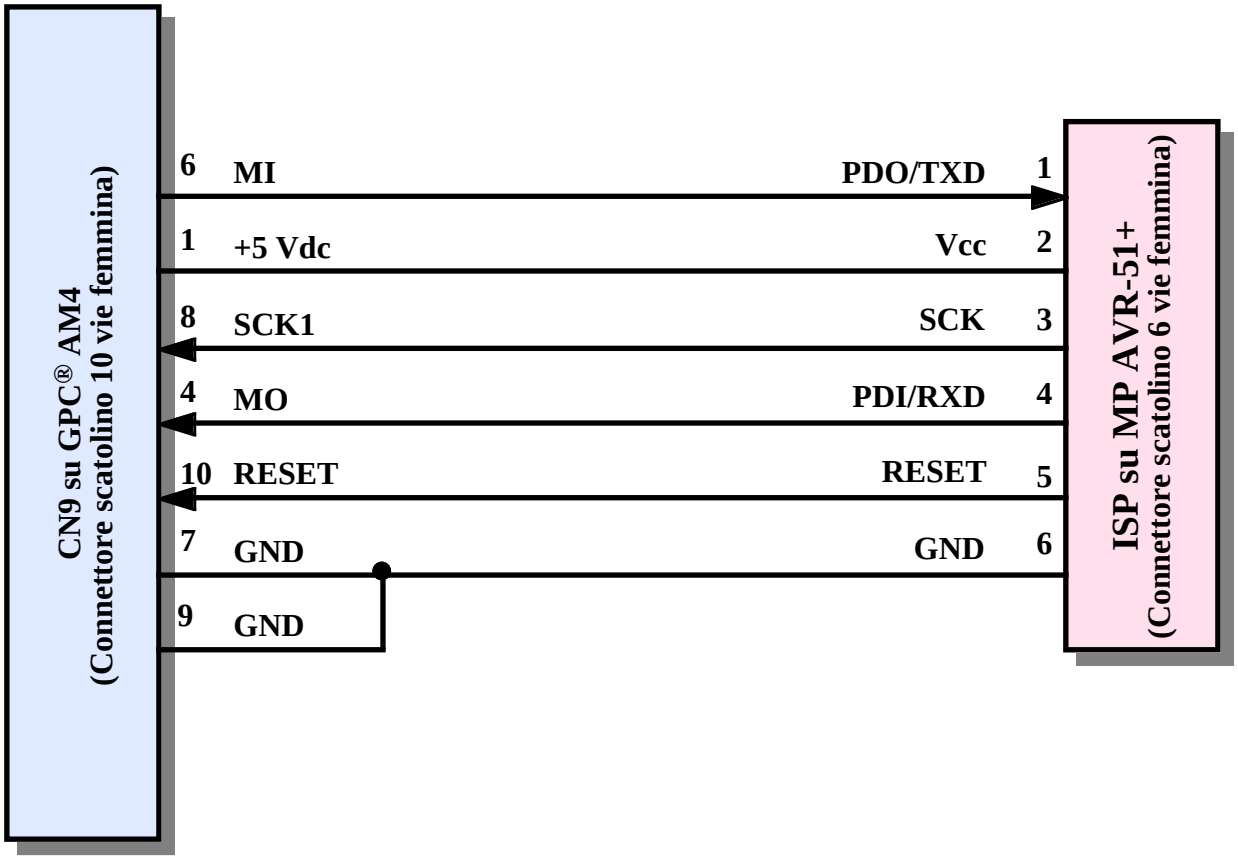


FIGURA 19: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE MP AVR-51+

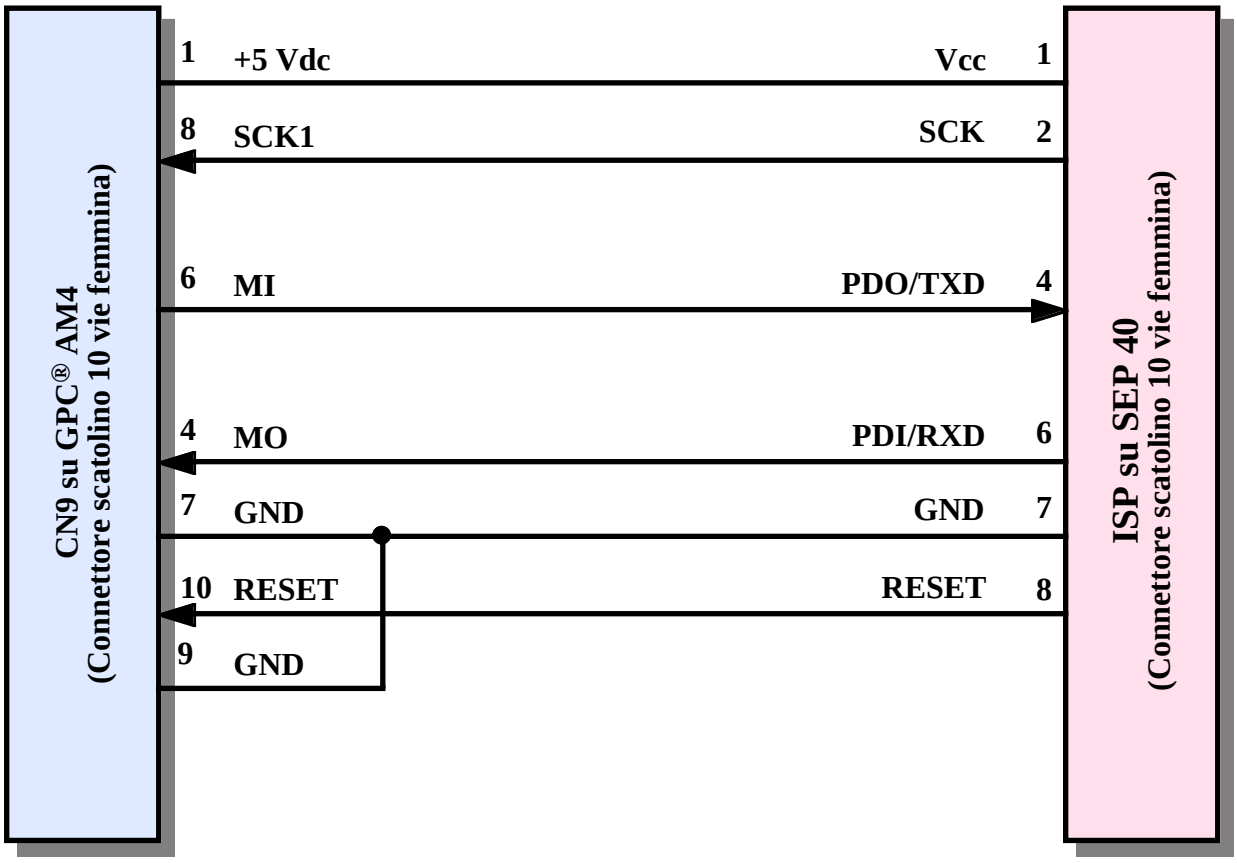


FIGURA 20: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE SEP 40

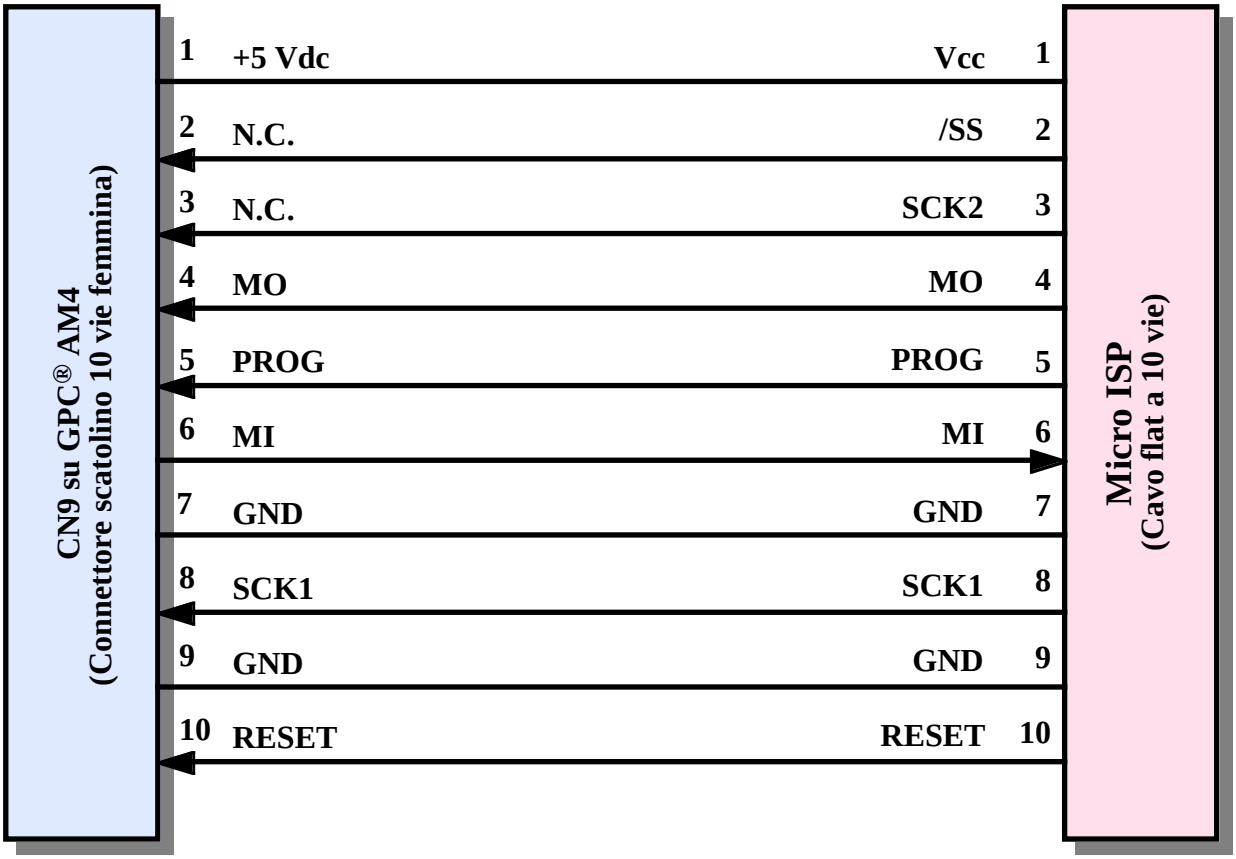


FIGURA 21: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE UISP-S4

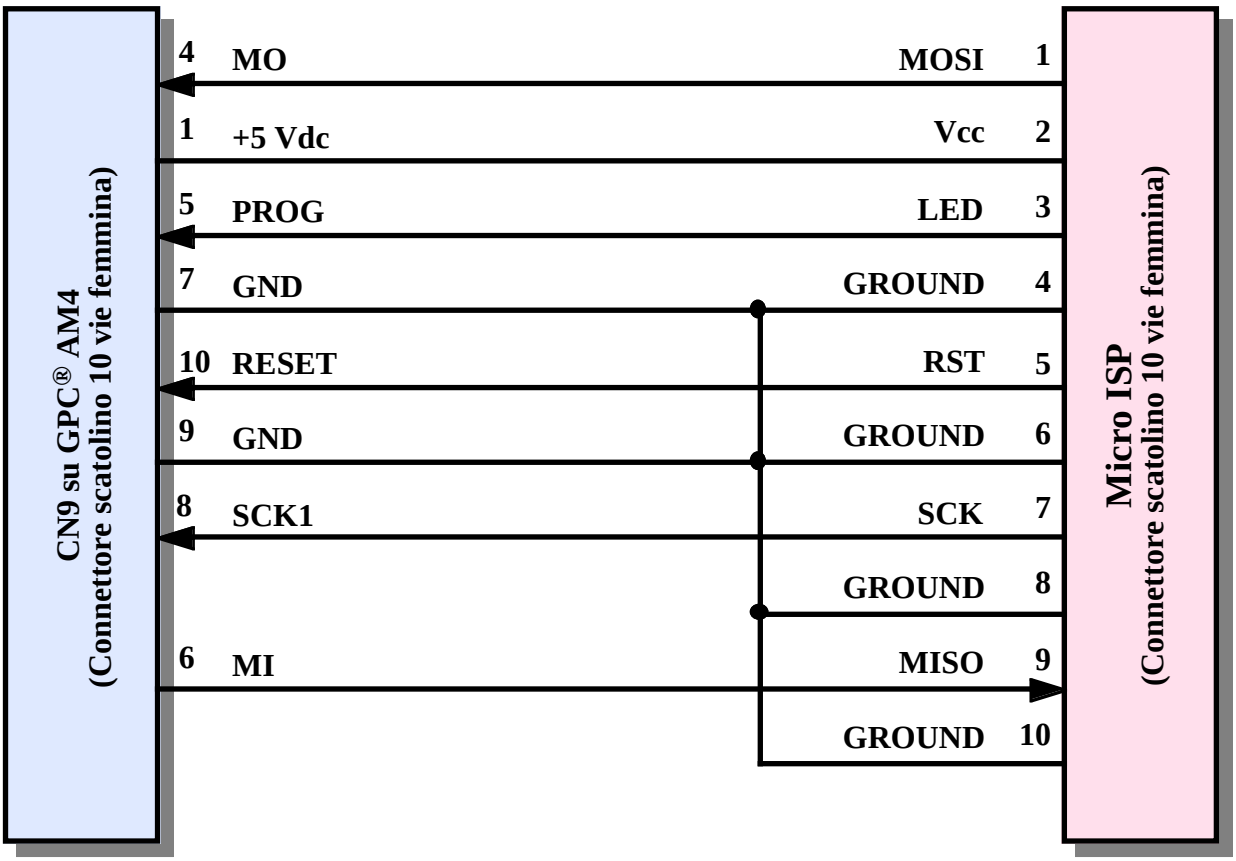


FIGURA 22: ESEMPIO COLLEGAMENTO ISP CON PROGRAMMATORE STK 300

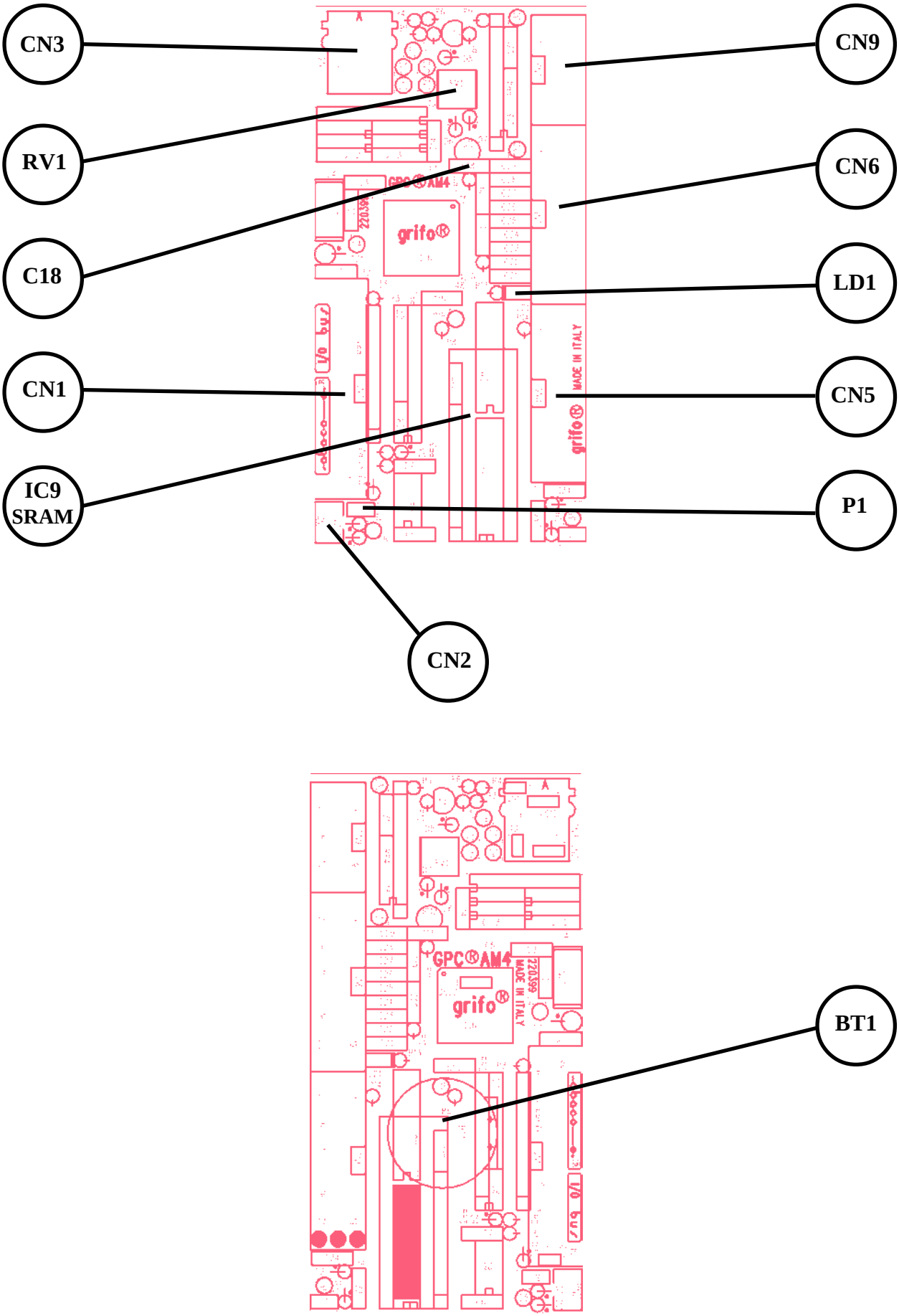


FIGURA 23: DISPOSIZIONE CONNETTORI, TRIMMER, BATTERIA, MEMORIE, ECC.

INTERFACCIAMENTO CONNETTORI CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui la **GPC® AM4** si deve interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Per i segnali che riguardano la comunicazione seriale con i protocolli RS 232, RS 422, RS 485 e current loop fare riferimento alle specifiche standard di ognuno di questi protocolli.
- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello 5V corrisponde allo stato logico 1.
- I segnali d'ingresso alla sezione A/D devono essere collegati a segnali analogici che rispettino il range di variazione ammesso: 0÷2,490 V. Da notare che gli 8 ingressi analogici presenti su CN6 sono dotati di condensatori di filtro che garantiscono una maggiore stabilità sul segnale acquisito, ma che allo stesso tempo abbassano la frequenza di taglio (1 MHz che é comunque superiore alla massima frequenza di acquisizione A/D=200 KHz).
- Per evitare problemi di pilotaggio, al connettore CN1 per **ABACO®** I/O BUS possono essere collegate fino ad un massimo di tre schede periferiche. Inoltre per minimizzare l'effetto di eventuali disturbi dal campo, la lunghezza del flat cable a 26 vie di interconnessione non dovrebbe superare i 250 mm di lunghezza.

TRIMMER E TARATURE

Sulla **GPC® AM4** é presente il trimmer RV1 utilizzato per la taratura della scheda; tale componente permette di fissare il valore della tensione di riferimento su cui si basa la sezione di A/D converter. La scheda viene sottoposta ad un accurato test di collaudo che provvede a verificare la funzionalità della stessa ed allo stesso tempo a tararla in tutte le sue parti. La taratura viene effettuata in laboratorio a temperatura costante di +20 gradi centigradi, seguendo la procedura di seguito descritta:

- Si effettua la taratura di precisione della Vref della sezione A/D ad un valore di 2,4900 V regolando il trimmer RV1, tramite un multimetro galvanicamente isolato a 5 cifre collegato ai capi del condensatore C18.
- Si verifica la corrispondenza tra segnale analogico fornito in ingresso e combinazione letta dalla sezione A/D converter. La verifica viene effettuata fornendo un segnale di verifica con un calibratore campione e controllando che la differenza tra la combinazione determinata dalla scheda e quella determinata in modo teorico, non superi la somma degli errori della sezione A/D (±2 punti).
- Si blocca il trimmer della scheda, opportunamente tarato, tramite vernice.

Le sezioni d'interfaccia analogica utilizzano componenti di alta precisione che vengono addirittura scelti in fase di montaggio, proprio per evitare lunghe e complicate procedure di taratura. Per questo una volta completato il test di collaudo e quindi la taratura, il trimmer RV1 viene bloccato, in modo da garantire una immunità della taratura anche ad eventuali sollecitazioni meccaniche (vibrazioni, spostamenti, ecc.). L'utente di norma non deve intervenire sulla taratura della scheda, ma se lo dovesse fare (a causa di derive termiche, derive del tempo, ecc.) deve rigorosamente seguire la procedura sopra illustrata.
La tensione di riferimento coincide con la tensione di fondo scala per gli ingressi analogici; questo valore non può essere cambiato quindi se devono essere acquisiti segnali analogici di ampiezza

superiore, questi dovranno essere ridotti esternamente, ad esempio usando un economico partitore a resistenze.
Per una facile individuazione di RV1 e C18 a bordo scheda, si faccia riferimento alla figura 23 mentre per ulteriori informazioni sulla sezione A/D e sul comparatore analogico (che non richiede tarature) leggere l'appendice B.

INTERFACCIE PER I/O DIGITALI

Tramite CN5 (connettore compatibili con standard di I/O **ABACO®**) si può collegare la **GPC® AM4** ai numerosi moduli del carteggio **grifo®** che riportano lo stesso pin out. Dal punto di vista dell'installazione, queste interfaccie richiedono solo un flat cable da 20 vie intestato con due connettori da 20 vie (FLT.20+20) con cui é possibile portare anche le alimentazioni, mentre dal punto di vista software la gestione é altrettanto semplice ed immediata, infatti i pacchetti software disponibili per la **GPC® AM4** sono provvisti di tutte le procedure necessarie. Quest'ultime per la maggioranza dei pacchetti software disponibili, coincidono con dei "driver software" o delle librerie aggiunti al linguaggio di programmazione, che consentono di utilizzare direttamente le istruzioni ad alto livello dello stesso linguaggio di programmazione e quindi tutta la loro potenza.
Di particolare interesse é la possibilità di collegare direttamente una serie di moduli come:

- **QTP 16P**, **QTP 24P**, **KDx x24**, **DEB 01**, ecc. con cui risolvere tutti i problemi di interfacciamento operatore locale. Questi moduli sono già dotati delle risorse necessarie per gestire un buon livello di colloquio uomo-macchina (includono infatti display alfanumerici, tastiera a matrice e LEDs di visualizzazione) ad una breve distanza dalla **GPC® AM4**. Dal punto di vista software i driver disponibili rendono utilizzabili le risorse dell'interfaccia operatore direttamente con le istruzioni ad alto livello per la gestione della console.

- **MCI 64** con cui risolvere tutti i problemi di salvataggio di grosse quantità di dati. Questo modulo é dotato di un connettore per memory card PCMCIA su cui possono essere inserite vari tipi di memory card (RAM, FLASH, ROM, ecc) nei vari size disponibili. Dal punto di vista software i driver disponibili coincidono con un completo file system e rendono utilizzabili le memory card direttamente con le istruzioni ad alto livello per la gestione dei files.

- **IAC 01**, **DEB 01** con cui gestire una stampante con interfaccia parallela CENTRONICS. Quest'ultima può essere collegata direttamente all'interfaccia, con un cavo standard, e quindi gestita con le istruzioni relative alla stampante del linguaggio di programmazione utilizzato.

- **RBO xx**, **TBO xx**, **XBI xx**, **OBI xx** con cui bufferare i segnali di I/O TTL nei confronti del campo. Con questi moduli i segnali di input vengono convertiti in ingressi optoisolati di tipo NPN o PNP, mentre i segnali di output vengono convertiti in uscite galvanicamente isolate a transistor o relé.

Per maggiori informazioni relative alle interfaccie per I/O digitali si veda il capitolo "SCHEDE ESTERNE" e la documentazione del software utilizzato.

SEGNALAZIONI VISIVE

La scheda **GPC® AM4** é dotata delle segnalazioni visive descritte nella seguente tabella:

LED	COLORE	FUNZIONE
LD1	Rosso	Indica lo stato della programmazione in system: é attivo quando l'ISP é abilitato e viceversa (stato segnale PROG su CN9).

FIGURA 24: TABELLA DELLE SEGNALAZIONI VISIVE

La funzione principale di questo LEDs é quella di fornire un'indicazione visiva dello stato della scheda, facilitando quindi le operazioni di debug e di verifica di funzionamento di tutto il sistema. Per una più facile individuazione di tale segnalazioni visiva, si faccia riferimento alla figura 23, mentre per ulteriori informazioni sulla sua attivazione si faccia riferimento al paragrafo "PROGRAMMAZIONE ISP".

JUMPERS

Esistono a bordo della **GPC® AM4** 7 jumpers, di cui 1 a cavalliere e 6 a stagno, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalità di connessione.

JUMPER	N° VIE	UTILIZZO
J1	2	Definisce stato dell'ingresso di configurazione (RUN o DEBUG).
JS1, JS2	2	Collegano circuiteria di terminazione e forzatura alla linea seriale in RS 422, RS 485.
JS3	3	Seleziona tipo di collegamento per il pin 1 di CN3.
JS14	2	Collega batteria di bordo BT1 a circuiteria di back up.
JS15	3	Seleziona la direzione ed il modo operativo per la linea seriale in RS 422, RS 485.
JS23	3	Definisce collegamento del segnale PE.6 INT6 della CPU.

FIGURA 25: TABELLA RIASSUNTIVA JUMPERS

Di seguito é riportata una descrizione tabellare delle possibili connessioni dei 7 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alle figure 31, 32 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione dei jumpers a bordo della scheda, si utilizzino invece la figura 26.
In tutte le seguenti tabelle l'***** indica la connessione di default, ovvero quella impostata in fase di collaudo, con cui la scheda viene fornita.

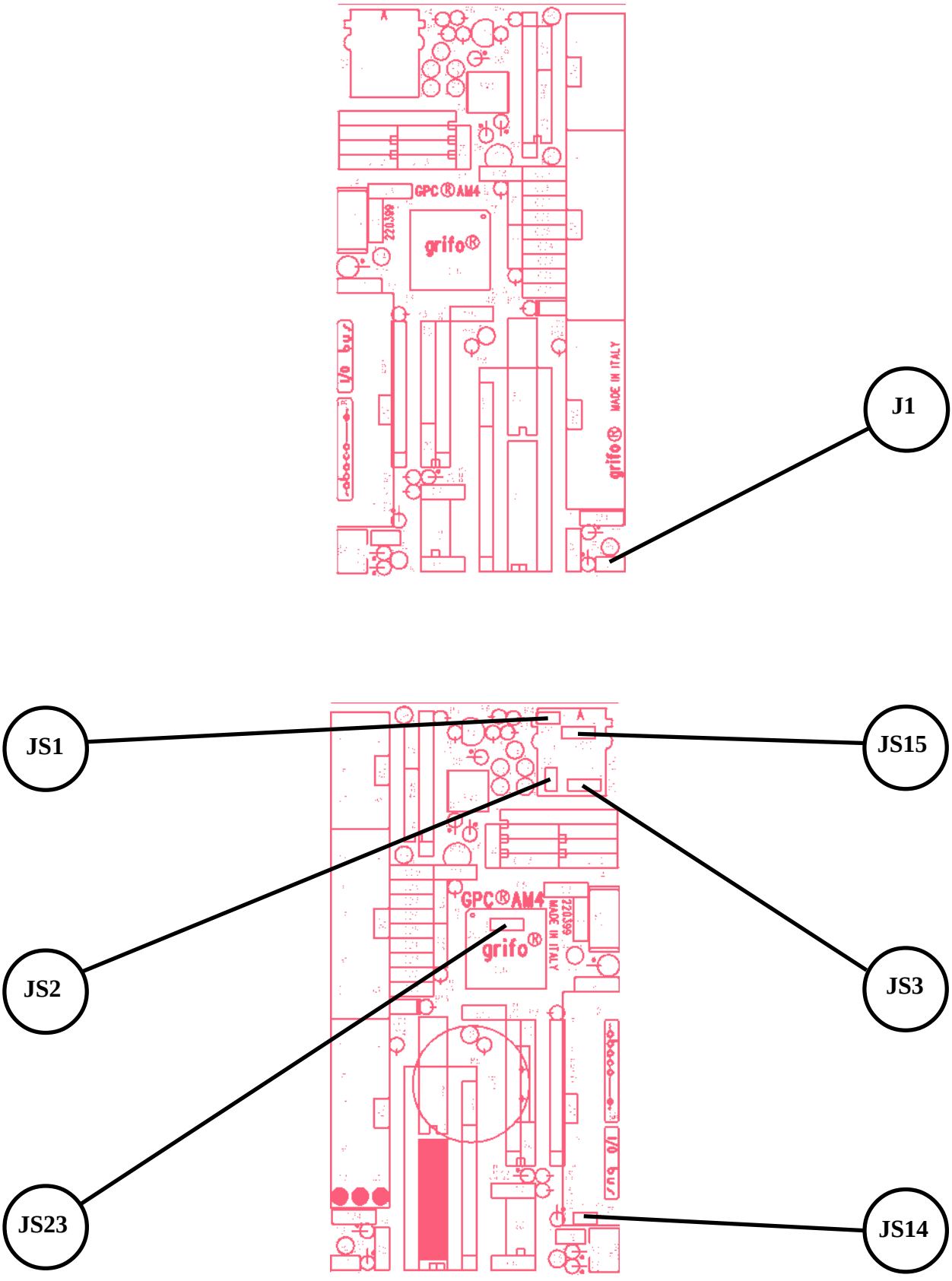


FIGURA 26: DISPOSIZIONE JUMPERS

JUMPERS A 2 VIE

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J1	non connesso	Collega l'ingresso di configurazione a +5Vdc e seleziona la modalità RUN.	*
	connesso	Collega l'ingresso di configurazione a GND e seleziona la modalità DEBUG.	
JS1, JS2	non connessi	Non collegano la circuiteria di terminazione e forzatura al ricevitore/trasmettitore RS 485 od al ricevitore RS 422, della linea seriale.	*
	connessi	Collegano la circuiteria di terminazione e forzatura al ricevitore/trasmettitore RS 485 od al ricevitore RS 422, della linea seriale.	
JS14	non connesso	Non collega batteria di bordo BT1 alla circuiteria di back up.	*
	connesso	Collega batteria di bordo BT1 alla circuiteria di back up.	

FIGURA 27: TABELLA JUMPERS A 2 VIE

JUMPERS A 3 VIE

JUMPER	CONNESSIONE	UTILIZZO	DEF.
JS3	posizione 1-2	Collega pin 1 di CN3 a GND.	*
	posizione 2-3	Collega pin 1 di CN3 a +5 Vdc.	
JS15	posizione 1-2	Configura la linea seriale per lo standard elettrico RS 485 (half duplex a 2 fili).	
	posizione 2-3	Configura la linea seriale A per lo standard elettrico RS 422 (full duplex o half duplex a 4 fili).	
JS23	posizione 1-2	Collega pin 8 della CPU (PE.6 INT6) al segnale di interrupt del real time clock.	*
	posizione 2-3	Collega pin 8 della CPU (PE.6 INT6) al segnale DIR di gestione abilitazione e direzione in RS 422 e RS 485.	

FIGURA 28: TABELLA JUMPERS A 3 VIE

JUMPERS A STAGNO

La connessione di default dei jumpers a stagno denominati **JSxx**, é effettuata con una sottile pista sul lato stagnature. Quindi, se tale configurazione deve essere variata, si deve prima tagliare la pista con un taglierino affilato e poi effettuare la connessione richiesta con uno stagnatore di bassa potenza, utilizzando dello stagno non corrosivo.

INTERRUPTS

Una caratteristica peculiare della **GPC® AM4** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di quali sono i dispositivi che possono generare interrupts e con quale modalità; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore oppure all'appendice B di questo manuale.

- **ABACO®** I/O BUS	->	Genera un INT7 sulla CPU, tramite la linea /INT BUS di CN1.
- Real Time Clock	->	Genera un INT6 sulla CPU.
- Port di I/O	->	Generano /INT0÷/INT3 ed INT4÷INT5 sulla CPU, tramite le ononime linee su CN5.
- Periferiche della CPU	->	Generano un interrupt interno. In particolare le possibili sorgenti d'interrupt interno sono le sezioni: Timer Counter, SPI, UART, ADC, Comparatore analogico, EEPROM.

Sulla scheda é presente un gestore d'interrupt che consente di attivare, disattivare, mascherare le sorgenti d'interrupt e che regolamenta l'attivazione contemporanea di più interrupts. Questa unità consente inoltre di definire via software gli indirizzi delle procedure di risposta agli interrupts, grazie ad un'apposita tabella di vettori salvata nella memoria del microprocessore. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

BACK UP

La **GPC® AM4** é provvista di una batteria al litio BT1 che provvede a tamponare la SRAM ed il Real Time Clock di bordo anche in assenza della tensione di alimentazione. Il jumper JS14 provvede a collegare o meno questa batteria in modo da salvaguardarne la durata prima dell'installazione o in tutti i casi in cui il back up non é necessario. Una seconda batteria esterna, con tensione superiore a quella di bordo, può essere collegata alla circuiteria di back up tramite il connettore CN2: quest'ultima non é interessata dalla configurazione del jumper JS14 e sostituisce a tutti gli effetti la BT1.
Per la scelta della batteria esterna di back up seguire le indicazioni del paragrafo "CARATTERISTICHE ELETTRICHE", per la sua individuazione si veda la figura 23.

SELEZIONE COMUNICAZIONE SERIALE

La linea di comunicazione seriale della scheda **GPC® AM4** può essere bufferata in RS 232, RS 422, RS 485 o current loop. Dal punto di vista software sulla linea può essere definito il protocollo fisico di comunicazione tramite la programmazione di alcuni registri interni della CPU. In dettaglio possono essere programmate per lavorare con 8,9 bit per carattere; 1 o 2 bit di stop; con baud rate standard e non standard, fino a 115200 Baud.
La selezione del protocollo elettrico avviene via hardware e richiede un'opportuna configurazione dei jumpers di bordo, come descritto nelle precedenti tabelle, e l'installazione di adeguati driver di comunicazione. Alcuni componenti necessari per le configurazioni RS 422, RS 485 e current loop non sono montati e collaudati sulla scheda in configurazione di default; per questo la prima configurazione della seriale non in RS 232 deve essere sempre effettuata dai tecnici **grifo®**. A questo punto l'utente può cambiare autonomamente la configurazione seguendo le informazioni sotto riportate:

- LINEA SERIALE SETTATA IN RS 232 (configurazione default)

			IC1	= driver MAX 202
JS1, JS2	=	non connessi	IC2	= nessun componente
JS15	=	indifferente	IC3	= nessun componente
JS23	=	indifferente	IC4	= nessun componente
			IC5	= nessun componente

- LINEA SERIALE SETTATA IN CURRENT LOOP (opzione .CLOOP)

			IC1	= nessun componente
JS1, JS2	=	non connessi	IC2	= nessun componente
JS15	=	indifferente	IC3	= nessun componente
JS23	=	indifferente	IC4	= driver HP 4200
			IC5	= driver HP 4100

Da ricordare che l'interfaccia seriale in current loop é di tipo passivo e si deve quindi collegare una linea current loop attiva, ovvero provvista di un proprio alimentatore come descritto nelle figure 11÷13. L'interfaccia current loop può essere utilizzata per realizzare sia connessioni punto punto che reti multipunto con un collegamento a 4 o 2 fili.

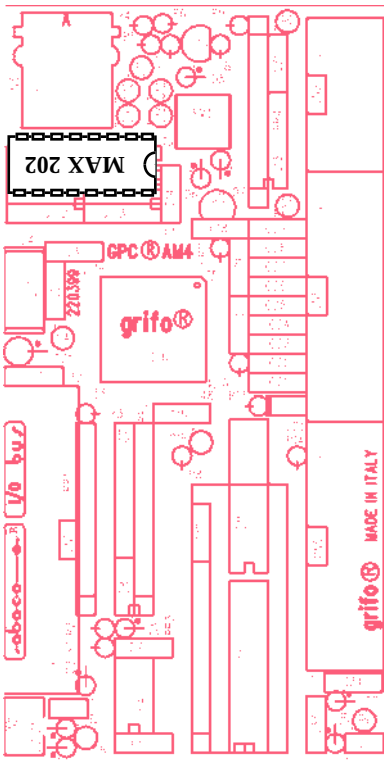
- LINEA SERIALE SETTATA IN RS 422 (opzione .RS 422)

			IC1	= nessun componente
JS1, JS2	=	(*)	IC2	= driver SN 75176 o MAX 483
JS15	=	posizione 2-3	IC3	= driver SN 75176 o MAX 483
JS23	=	posizione 2-3	IC4	= nessun componente
			IC5	= nessun componente

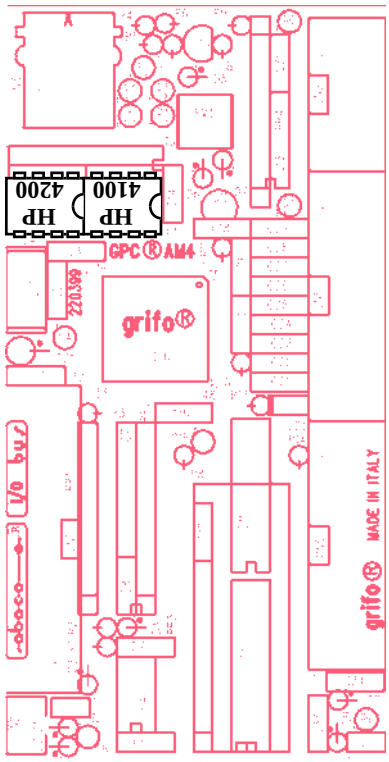
Lo stato del segnale DIR=PE.6 (gestito via software con i registri del Port E) consente di abilitare o disabilitare il trasmettitore come segue:

DIR=PE.6	= livello basso	= stato logico 0	->	trasmettitore attivo
DIR=PE.6	= livello alto	= stato logico 1	->	trasmettitore disattivo

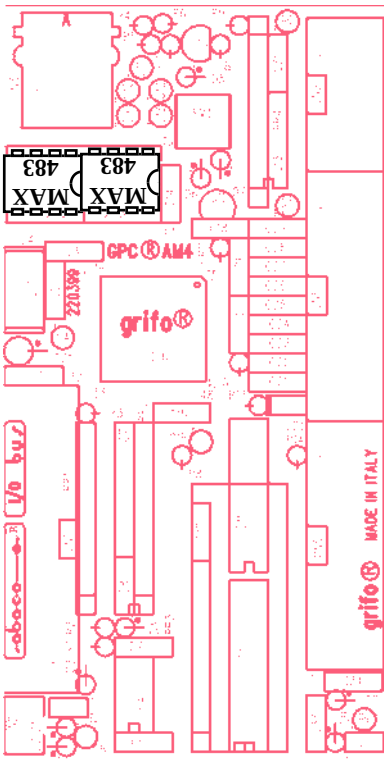
Per sistemi punto punto, la linea DIR=PE.6 può essere mantenuta sempre bassa (trasmettitore sempre attivo), mentre per reti multipunto si deve attivare il trasmettitore solo in corrispondenza della trasmissione. La comunicazione RS 422 é di tipo full duplex.



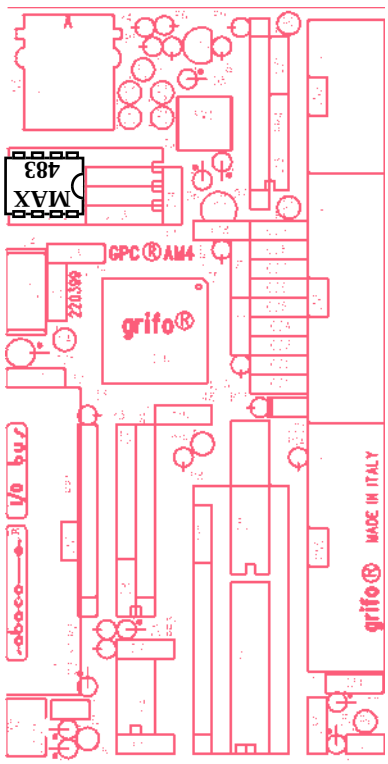
Seriale in RS 232



Seriale in current loop



Seriale in RS 422



Seriale in RS 485

FIGURA 29: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE

- LINEA SERIALE SETTATA IN RS 485 (opzione .RS 485)

			IC1	=	nessun componente
JS1, JS2	=	(*)	IC2	=	driver SN 75176 o MAX 483
JS15	=	posizione 1-2	IC3	=	nessun componente
JS23	=	posizione 2-3	IC4	=	nessun componente
			IC5	=	nessun componente

In questa modalità le linee da utilizzare sono i pin 4 e 5 di CN3, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato del segnale DIR=PE.6 (gestito via software con i registri del Port E) come segue:

DIR=PE.6	= livello basso	= stato logico 0	->	linea in trasmissione
DIR=PE.6	= livello alto	= stato logico 1	->	linea in ricezione

Questa comunicazione la si utilizza sia per connessioni punto punto che multipunto con una comunicazione half duplex. Sempre in questa modalità si riceve quanto trasmesso, in modo da fornire al sistema la possibilità di verificare autonomamente la riuscita della trasmissione; infatti in caso di conflitti sulla linea, quanto trasmesso non viene ricevuto correttamente e viceversa.

(*) Nel caso si utilizzi la linea seriale in RS 422 o RS 485, con i jumpers JS1 e JS2 é possibile connettere la circuiteria di terminazione e forzatura sulla linea . Tale circuiteria deve essere sempre presente nel caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.
In fase di reset o power on, il segnale DIR=PE.6 è mantenuto a livello logico alto di conseguenza in seguito ad una di queste fasi il driver RS 485 è in ricezione o il driver di trasmissione RS 422 è disattivo, in modo da eliminare eventuali conflittualità sulla linea di comunicazione.

Per ulteriori informazioni relative alla comunicazione seriale fare riferimento agli esempi di collegamento delle figure 7÷13 ed all'appendice B.

PIN MULTIFUNZIONE

Il microcontrollore AVRmega103 usato a bordo della **GPC® AM4** é provvisto di sei port ad 8 bit di I/O generico per un totale di 48 linee. Molte di queste linee sono fisicamente multiplexate all'interno dello stesso microprocessore e possono quindi assumere funzionalità diverse a seconda della programmazione software effettuata. Nella seguente tabella viene riportato l'elenco di tutti i segnali che hanno molteplici funzioni con il relativo uso a bordo scheda e lo stato assunto durante la fase di reset e power on.

SEGNALE	FUNZIONE MPX	USO SU GPC® AM4	STATO AL RESET
PA.0÷PA.7	AD0÷AD7	A0÷A7, D0÷D7	Tri state
PB.0	/SS	PB.0, /SS su CN5	Tri state
PB.1	SCK	PB.1, SCK su CN5; SCK1 su CN9	Tri state
PB.2	MOSI	PB.2, MOSI su CN5	Tri state
PB.3	MISO	PB.3, MISO su CN5	Tri state
PB.4	OC0, PWM0	PB.4, OC0, PWM0 su CN5	Tri state
PB.5	OC1A, PWM1A	PB.5, OC1A, PWM1A su CN6	Tri state
PB.6	OC1B, PWM1B	PB.6, OC1B, PWM1B su CN6	Tri state
PB.7	OC2, PWM2	PB.7, PWM2 su CN5	Tri state
PC.0÷7	A8÷A15	A8÷A15	Uscite basse (0 V)
PD.0	/INT0	PD.0, /INT0 su CN5	Tri state
PD.1	/INT1	PD.1, /INT1 su CN5	Tri state
PD.2	/INT2	PD.2, /INT2 su CN5	Tri state
PD.3	/INT3	PD.3, /INT3 su CN5	Tri state
PD.4	IC1	PD.4, IC1 su CN5	Tri state
PD.5	-	PD.5 su CN5	Tri state
PD.6	T1	PD.6, T1 su CN5	Tri state
PD.7	T2	PD.7, T2 su CN5	Tri state
PE.0	PDI, RXD	RX .... su CN3; MO su CN9	Tri state
PE.1	PDO, TXD	TX .... su CN3; MI su CN9	Tri state
PE.2	AC+	PE.2, AC+ su CN5	Tri state
PE.3	AC-	PE.3, AC- su CN5	Tri state
PE.4	INT4	PE.4, /INT4 su CN5	Tri state
PE.5	INT5	PE.5, /INT5 su CN5	Tri state
PE.6	INT6	/INT RTC, DIR	Tri state
PE.7	INT7	/INT BUS su CN1	Tri state
PF.0÷PF.7	ADC0÷ADC7	ADC0÷ADC7 su CN6	Ingressi ad alta impedenza

FIGURA 30: TABELLA PIN MULTIFUNZIONE

L'utente può arbitrariamente decidere il tipo di funzionalità dei segnali collegati ai connettori della scheda, mentre deve programmare con scrupolosa attenzione i segnali dedicati alla gestione della memoria esterna (Port A e Port C). Normalmente i pacchetti software sviluppati per la scheda si preoccupano già di effettuare il giusto settaggio di tutte le linee dei Port ed all'utente rimane solo da acquisire il precedente settaggio, cambiare i segnali ammessi e settare la nuova funzionalità.
Come descritto nella figure 15 e 17 le linee multifunzione della CPU sono collegate ai connettori della scheda direttamente senza alcuna resistenza di pull up o pull down e visto che durante il reset tutte queste linee sono in tree state, si ricava che il loro stato durante questa fase é stabilito esclusivamente dalla circuiteria esterna collegata.

RESET E WATCH DOG

La scheda **GPC® AM4** è dotata di una circuiteria di watch dog molto efficiente e di facile gestione software. In particolare le caratteristiche di questa circuiteria sono le seguenti:

- funzionamento astabile;
- tempo d'intervento programmabile da 12 a 1600 msec;
- attivazione via software;
- disattivazione via software protetta;
- retrigger via software.

Si ricorda che nel funzionamento astabile una volta scaduto il tempo d'intervento la circuiteria si attiva, rimane attiva per il tempo di reset (circa 130 msec) e quindi si disattiva autonomamente. Le operazioni di retrigger, abilitazione e disabilitazione del watch dog sono effettuate via software sugli appositi registri interni descritti nell'appendice B.
In corrispondenza dell'attivazione e sucessiva disattivazione del segnale di /RESET la scheda riprende l'esecuzione del programma salvato sulla FLASH interna della CPU (all'indirizzo 0000H), partendo da una condizione di azzeramento generale.
Si ricorda inoltre che il segnale di /RESET generato dalla scheda é collegato al pin 20 del connettore CN1 e che tra le sorgenti di reset della **GPC® AM4**, oltre all'eventuale circuiteria di watch dog, sono sempre presenti la circuiteria di power good, l'interfaccia ISP ed il contatto di reset **P1**. I due pin di P1 possono essere collegati ad un contatto normalmente aperto (es.un pulsante) e quando il contatto é chiuso (cortocircuito dei due pin) si attiva la circuiteria di /RESET con le dovute modalità e tempistiche. La sua funzione principale é quella di uscire da condizioni di loop infinito, soprattutto durante la fase di debug oppure per rieseguire il programma applicativo senza dover interrompere l'alimentazione della scheda. Per una facile individuazione di tale contatto a bordo scheda, si faccia riferimento alla figura 23.
Si ricorda che un reset generato dalla circuiteria di watch dog non abilita il segnale /RESET su CN1, quindi le eventuali periferiche collegate all'**ABACO®** I/O BUS non sono resettate. In ogni modo l'utente può riconoscere l'avvenuto reset da watch dog testando il registro di stato MCUSR interno alla CPU, all'inizio del suo programma applicativo. Se il reset appena avvenuto é dovuto al watch dog lo stesso programma applicativo potrà inizializzare opportunamente tutte le schede periferiche.

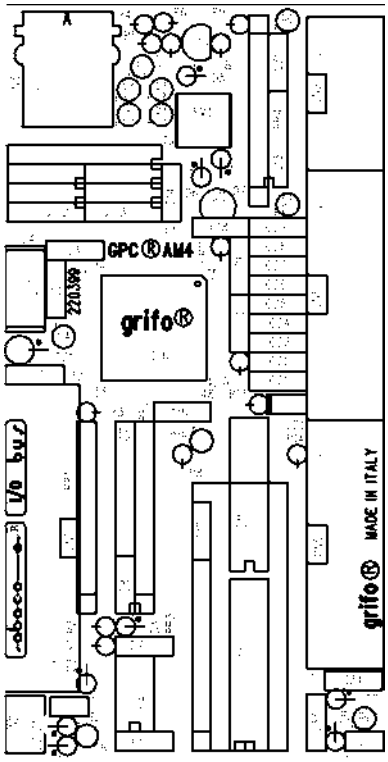


FIGURA 31: PIANTA COMPONENTI LATO COMPONENTI

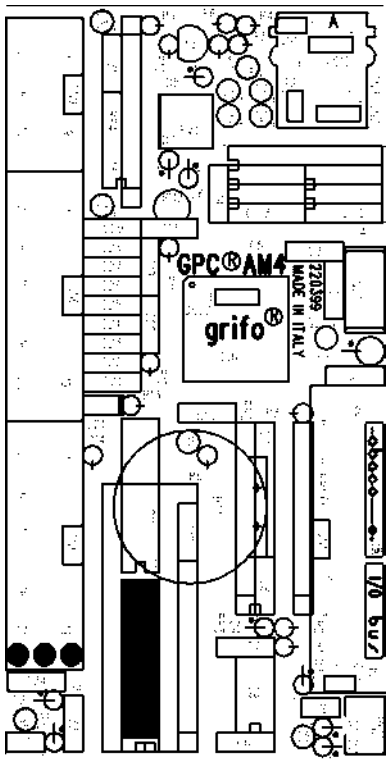


FIGURA 32: PIANTA COMPONENTI LATO STAGNATURE

PROGRAMMAZIONE ISP

Una delle caratteristiche più importanti della **GPC® AM4** é la possibilità di usare la programmazione in system, ovvero la programmazione effettuata a bordo scheda, senza dover togliere la CPU. Di seguito sono descritti i passi da effettuareper un normale sviluppo di programma applicativo funzionante:

1) collegare il programmatore ISP all'apposito connettore CN9, come descritto nelle figure 18÷22;
2) alimentare la scheda;
3) predisporre il il programmatore ISP per il microprocessore usato (ATmega103) e per gli eventuali altri settaggi necessari usando l'apposito software di gestione ISP;
4) sviluppare il programma applicativo tramite un pacchetto software che generi un codice eseguibile;
5) programmare la FLASH EPROM interna del microprocessore con il programma applicativo generato al punto 4, usando l'apposito software di gestione ISP del punto 3;
6) avviare l'esecuzione del programma applicativo salvato al punto 5 e verificarne quindi il funzionamento;
7) in caso di funzionamento non corretto ripetere i passi 4÷6 fino alla completa messa a punto del programma applicativo, viceversa passare al passo sucessivo;
8) togliere alimentazione alla scheda;
9) scollegare programmatore ISP da CN9;
10) rialimentare la scheda: il programma applicativo é eseguito autonomamente dalla FLASH ROM interna.

L'ISP riduce i costi complessivi dell'applicazione, infatti elimina l'uso di EPROM, programmatore di EPROM, FLASH EPROM, ecc. Per ulteriori informazioni relative alla programmazione ISP fare riferimento alla specifica documentazione tecnica della ATMEL.

DESCRIZIONE SOFTWARE

Questa scheda ha la possibilità di usufruire di una ricca serie di strutture software che consentono di utilizzarne il modulo come sistema di sviluppo di sé stesso, sia in assembler che in linguaggi ad alto livello. In questo modo l'utente può facilmente sviluppare le applicazioni necessarie in un tempo veramente corto. In generale la scheda può utilizzare tutte le risorse software disponibili per il processore montato, ovvero i numerosi pacchetti ideati per l'AVR e derivati.
Tutti i pacchetti di sviluppo software forniti dalla **grifo®** sono sempre accompagnati da esempi che illustrano come gestire ogni sezione della scheda e da una completa documentazione d'uso.
Tra questi ricordiamo:

BASCOM AVR
Cross compilatore a basso costo per files sorgenti scritti in BASIC, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore ed un simulatore molto potente per il debugger del sorgente. Comprende molti modelli di memoria, svariati tipi di dati, istruzioni dedicate alle risorse hardware e librerie aggiuntive per simulare od interfacciare dispositivi intelligenti esterni (tastiere PC, display alfanumerici e grafici, lettori di badge, ecc.). L'IDE é in grado di eseguire anche programmi esterni per la programmazione ISP.
Viene fornito su CD con la relativa manualistica tecnica e gli esempi d'uso.

ICC AVR PRO
Cross compilatore per files sorgenti scritti in ANSI C, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore un ottimizzatore ed un linker. Comprende molti modelli di memoria, svariati tipi di dati, le classiche librerie ANSI con i relativi sorgenti, gestione completa del floating point, utilità, ecc. Il compilatore si integra con il pacchetto AVR Studio con cui si può effettuare il debugger a livello sorgente dell'applicativo utente in C e con programmi esterni per la programmazione ISP.
Viene fornito su CD con la relativa manualistica tecnica e gli esempi d'uso.

DDS MICRO C 86
E’ un comodo pacchetto software, a basso costo, che tramite un completo IDE permette di utilizzare un editor, un compilatore “C” (integer), un assemblatore, un linker ed un ottimizzatoree. Sono inclusi i sorgenti delle librerie, una serie di utility ed una ricca documentazione su dischetto da 3”1/2 nel formato MS-DOS.

AVR Studio
E' un ambiente di sviluppo per la famiglia AVR di microprocessori che consente di controllare l'esecuzione dei programmi applicativi tramite appositi in circuit emulator esterni oppure in simulazione su PC. AVR Studio consente il debug a livello sorgente assembly e/o C di codici oggetto generati da pacchetti esterni (assemblatori, compilatori, ecc.) e si presenta con una serie di finestre che riportano le principali informazioni necessarie come i registri, la memoria, le periferiche, le variabili in modo da fornire all'utente il pieno controllo di ogni elemento. Include inoltre un "costruttore di applicativo" che facilita la generazione del codice che inizializza le varie periferiche hardware (UART, SPI, Port, ADC, ecc) partendo da un'interfaccia grafica.

MAPPAGGI ED INDIRIZZAMENTI

In questo capitolo ci occuperemo di fornire tutte le informazioni sulle caratteristiche hardware della scheda relative al mappaggio delle memorie, delle periferiche, dei registri, ecc. necessarie per la gestione software delle sezioni componenti.

MAPPAGGIO DELLE RISORSE DI BORDO

La gestione delle risorse della scheda è interamente affidata ad una specifica logica di controllo combinatoria, realizzata con porte CMOS. Questa logica alloca la memoria ed i dispositivi periferici esterni alla CPU nel suo spazio d'indirizzamento con un minimo assorbimento di corrente ed una semplice gestione software.
Complessivamente la CPU ATmega103 indirizza direttamente 64K words di memoria codice e di 64K bytes di memoria dati e la logica di controllo alloca i dispositivi esterni all'interno di quest'ultimo spazio.
Riassumendo i seguenti dispositivi sono allocati sulla **GPC® AM4**:

- **ABACO®** I/O BUS
- 32K di SRAM su IC 9
- Ingresso di configurazione J1
- Real Time Clock
- Memorie interne al microprocessore
- Periferiche interne al microprocessore

La gestione del mappaggio é realizzata via hardware quindi gli indirizzi dei dispositivi elencati non può essere variata con valori diversi. Se sono necessari mappaggi particolari e/o diversi si prega di contattare direttamente la **grifo®**. Informazioni dettagliate sul mappaggio ed indirizzamenti sono riportati nei paragrafi seguenti e nell'appendice B.

MAPPAGGIO ABACO® I/O BUS

La logica di controllo della **GPC® AM4** provvede anche alla gestione dell'**ABACO®** I/O BUS, definendo gli indirizzi in cui tale BUS viene allocato; come si può notare dalla sucessiva figura 35, tale BUS è indirizzato in corrispondenza degli indirizzi 8000H÷80FFH per una estensione di 256 bytes. Un accesso in I/O in un qualsiasi indirizzo compreso in questo range abilita il segnale /IORQ e tutti gli altri segnali di controllo di CN1. Nella fase di mappaggio di eventuali schede periferiche collegate alla scheda, solo il byte meno significativo dell'indirizzo di I/O é significativo in quanto l'**ABACO®** I/O BUS prevede solo 8 bit d'indirizzamento ed 8 bit di dati.

MAPPAGGIO MEMORIE

Gli indirizzi di memoria sulla sono descritti nella seguente figura:

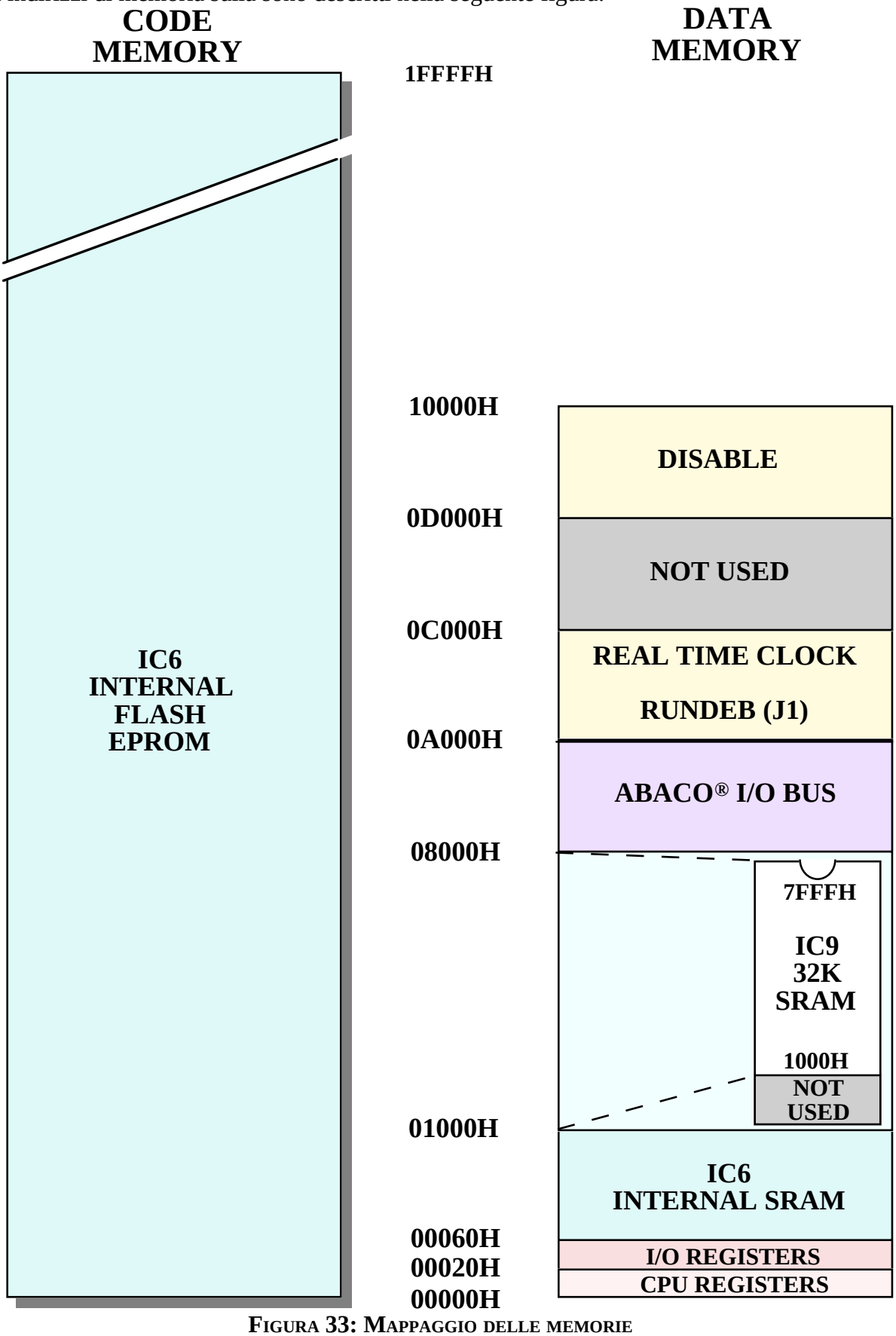


FIGURA 33: MAPPAGGIO DELLE MEMORIE

I 164K bytes massimi di memoria sono allocati sulla scheda come di seguito descritto:

- 64K words oppure 128K bytes di FLASH EPROM interna allocata nella memoria codice;
- 4K bytes di SRAM interna allocata nella memoria dati;
- 4K bytes di EEPROM interna;
- 32K bytes di SRAM esterna allocata nella memoria dati: di questo dispositivo solo gli ultimi 28K bytes sono indirizzabili dalla CPU mentre i rimanenti primi 4K bytes non sono utilizzati.

Il suffisso interna significa che il dispositivo é fisicamente presente all'interno del microprocessore mentre la memoria codice e dati sono le aree indirizzabili dalla CPU con apposite e diverse modalità. In dettaglio sulla **GPC® AM4** si può accedere ai diversi dispositivi di memoria, come descritto nella seguente tabella:

MEMORIA	OPERAZIONE LETTURA	OPERAZIONE SCRITTURA
FLASH EPROM INTERNA	Interfaccia ISP Istruzioni dirette della CPU	Interfaccia ISP
REGISTRI INTERNI	Istruzioni dirette della CPU	Istruzioni dirette della CPU
SRAM INTERNA	Istruzioni dirette della CPU	Istruzioni dirette della CPU
EEPROM INTERNA	Interfaccia ISP Istruzioni indirette della CPU	Interfaccia ISP Istruzioni indirette della CPU
SRAM ESTERNA	Istruzioni dirette della CPU	Istruzioni dirette della CPU
REGISTRI ESTERNI	Istruzioni dirette della CPU	Istruzioni dirette della CPU

FIGURA 34: ACCESSO MEMORIE

La SRAM esterna ed i registri esterni possono essere usati tramite le istruzioni dirette della CPU solo se l'accesso alla memoria dati esterna é stato abilitato tramite l'apposito registro interno MCUCR. A seguito di un reset o power on, la scheda inizia l'esecuzione del codice salvato all'indirizzo **00000H** della FLASH EPROM interna, con l'accesso alla memoria dati esterna **disabilitato**.
Per ulteriori informazioni su memorie, modalità d'indirizzamento, tempistiche, ecc. fare riferimento all'appendice B.

MAPPAGGIO PERIFERICHE

La logica di controllo della scheda alloca i dispositivi periferici esterni negli ultimi 32K Bytes dell'area dati della CPU. Per maggior chiarezza nella seguente tabella si riportano i nomi, gli indirizzi, i tipi di accesso ed una breve descrizione del significato, di tutti i registri delle periferiche esterne.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
ABACO® I/O BUS	I/OBUS	8000H÷80FFH	R/W	Indirizzi gestione **ABACO®** I/O BUS
J1	RUNDEB	A000H	R	Registro acquisizione stato dell'ingresso di configurazione J1
REAL TIME CLOCK 62421	SEC1	A000H	R/W	Registro dati per unità secondi
	SEC10	A001H	R/W	Registro dati per decine secondi
	MIN1	A002H	R/W	Registro dati per unità minuti
	MIN10	A003H	R/W	Registro dati per decine minuti
	HOU1	A004H	R/W	Registro dati per unità ore
	HOU10	A005H	R/W	Registro dati per decine ore e AM/PM
	DAY1	A006H	R/W	Registro dati per unità giorno
	DAY10	A007H	R/W	Registro dati per decine giorno
	MON1	A008H	R/W	Registro dati per unità mese
	MON10	A009H	R/W	Registro dati per decine mese
	YEA1	A00AH	R/W	Registro dati per unità anno
	YEA10	A00BH	R/W	Registro dati per decine anno
	WEE	A00CH	R/W	Registro dati per giorno della settimana
	REGD	A00DH	R/W	Registro di controllo D
	REGE	A00EH	R/W	Registro di controllo E
	REGF	A00FH	R/W	Registro di controllo F
SEGNALI CONTR.	DIS	E000H	R/W	Registro disabilitazione segnali di controllo

FIGURA 35: TABELLA INDIRIZZAMENTO PERIFERICHE ESTERNE

Per quanto riguarda la descrizione dettagliata dei registri sopra riportati, si faccia riferimento al capitolo sucessivo “DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO”.
Si ricorda che la figura 35 riporta la descrizione dei soli registri delle periferiche esterne al microprocessore e che per la descrizione di quelli interni si può fare riferimento all’apposita documentazione della casa costruttrice o all'APPENDICE B.
La logica di controllo elimina ogni conflitto tra le periferiche interne ed esterne, mantenendo ben separati i relativi registri di gestione.

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Nel paragrafo precedente sono stati riportati gli indirizzi di allocazione di tutte le periferiche esterne e di seguito viene riportata una descrizione dettagliata della funzione e del significato dei relativi registri e o linee (al fine di comprendere le successive informazioni, fare sempre riferimento alle tabelle delle figure 34 e 35). Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. In questo paragrafo inoltre non vengono descritte le sezioni che fanno parte del microprocessore; per quanto riguarda la programmazione di quest'ultime si faccia riferimento all'appendice B di questo manuale. Nei paragrafi successivi si usano le indicazioni **D0÷D7** e **.0÷7** per fare riferimento ai bits della combinazione utilizzata nelle operazioni di I/O.

INGRESSO DI CONFIGURAZIONE

La **GPC® AM4** dispone di un jumper di configurazione settabile dall'utente ed acquisibile via software, con le modalità di seguito riportate. Le possibili applicazioni di questi ingressi possono essere quelle destinate al settaggio di alcune condizioni di lavoro, alla selezione di parametri relativi al firmware di bordo, selezione della lingua, ecc.
Lo stato del jumper di configurazione J1 può essere accquisito via software, effettuando una semplice operazione di lettura da area dati, all'indirizzo del registro **RUNDEB** ed esaminando il bit D7. L'acquisizione è in logica negata, ovvero:

J1 connesso -> RUNDEB.7 = 0
J1 non connesso -> RUNDEB.7 = 1

Tale jumper può svolgere anche la funzione di selettore delle modalità RUN (non connesso) o DEBUG (connesso), caratteristica di alcuni pacchetti software della **grifo®**. Il registro RUNDEB condivide l'indirizzo di altre periferiche di bordo ma questo non genera conflitti infatti l'acquisizione del jumper é un'operazione di lettura ed il bit D7 non é usato da questi altri dispositivi.
Per una facile individuazione dell'ingresso di configurazione sulla scheda, si veda la figura 26.

DISABILITAZIONE SEGNALI DI CONTROLLO

Il registro **DIS** può essere utilizzato per disabilitare i segnali di controllo esterni a seguito di ogni accesso agli altri registri esterni, infatti il microprocessore della **GPC® AM4** mantiene settati i segnali di indirizzo, usati dalla logica di controllo per generare appunto i segnali di controllo delle periferiche esterne, anche quando l'operazione di lettura e scrittura é terminata. I segnali di controllo che rimangono abilitati sono solo i chip select mentre gli altri segnali di controllo (/RD e /WR) sono correttamente disabilitati. Se l'utente desidera mantenere tutti i segnali di controllo normalmente disabilitati, può semplicemente effettuare un'operazione di lettura o scrittura sul registro DIS a seguito di ogni accesso agli altri registri esterni. Il dato letto o scritto sul registro DIS é privo di significato e può essere trattato arbitrariamente dall'utente.
Per i dispositivi periferici di bordo (real time clock ed ingresso di configurazione) l'operazione di disabilitazione non é necessaria mentre é indispensabile dopo ogni aceeso all'**ABACO®** I/O BUS, in particolare quando l'elettronica collegata a CN1 é sviluppata dall'utente ed é basata sul segnale di controllo /IORQ.

REAL TIME CLOCK

Questa periferica è vista in 16 registri consecutivi di cui 3 di stato e le rimanenti 13 per i dati. I registri dati sono utilizzati sia per operazioni di lettura (dell'orario attuale) che di scrittura (per il settaggio dell'orologio) così come i registri di stato e controllo i quali sono utilizzati in scrittura (per la programmazione del modo di funzionamento dell'orologio) ed in lettura (per determinare lo stato dell'orologio).
Per quanto riguarda il significato dei registri dati vale la corrispondenza:

SEC1	- Unita' dei secondi	- 4 bit meno significativi:	**SEC1.3÷SEC1.0**
SEC10	- Decine dei secondi	- 3 bit meno significativi:	**SEC10.2÷SEC10.0**
MIN1	- Unita' dei minuti	- 4 bit meno significativi:	**MIN1.3÷MIN1.0**
MIN10	- Decine dei minuti	- 3 bit meno significativi:	**MIN10.2÷MIN10.0**
HOU1	- Unita' delle ore	- 4 bit meno significativi:	**HOU1.3÷HOU1.0**
HOU10	- Decine delle ore	- 2 bit meno significativi:	**HOU10.1÷HOU10.0**
		Il terzo bit di tale registro, **HOU10.2**, indica l'AM/PM	
DAY1	- Unita' del giorno	- 4 bit meno significativi:	**DAY1.3÷DAY1.0**
DAY10	- Decine del giorno	- 2 bit meno significativi:	**DAY10.1÷DAY10.0**
MON1	- Unita' del mese	- 4 bit meno significativi:	**MON1.3÷MON1.0**
MON10	- Decine del mese	- 1 bit meno significativo:	**MON10.0**
YEA1	- Unita' dell'anno	- 4 bit meno significativi:	**YEA1.3÷YEA1.0**
YEA10	- Decine dell'anno	- 4 bit meno significativi:	**YEA10.3÷YEA10.0**
WEE	- Giorno della settimana	- 3 bit meno significativi:	**WEE.2÷WEE.0**

Per quest'ultimo registro vale la corrispondenza:

WEE.2	**WEE.1**	**WEE.0**	Giorno della settimana
0	0	0	Domenica
0	0	1	Lunedi'
0	1	0	Martedi'
0	1	1	Mercoledi'
1	0	0	Giovedi'
1	0	1	Venerdi'
1	1	0	Sabato

I tre registri di controllo sono invece utilizzati come segue:

	D7	D6	D5	D4	D3	D2	D1	D0
REGD =	**NU**	**NU**	**NU**	**NU**	**30S**	**IF**	**B**	**H**

dove:

NU = Non usato

30S = Se attivo (1) permette di effettuare una correzione di 30 secondi dell'orario. Una volta settato i secondi del RTC vengono azzerati ed i minuti incrementati se il precedente valore dei secondi era superiore o uguale a 30.

IF = Gestisce lo stato d'interrupt del RTC. In lettura riporta lo stato attuale d'interrupt (1=attivo e viceversa), mentre se resettato con una scrittura determina la fine interrupt, quando il RTC lavora in interrupt mode.

B = Indica se possono essere effettuate operazioni di lettura/scrittura dei registri:
1 -> operazioni impossibili e viceversa.

H = Se attivo (1) effettua la memorizzazione dell'orario fissato.

	D7	D6	D5	D4	D3	D2	D1	D0
REGE =	**NU**	**NU**	**NU**	**NU**	**T1**	**T0**	**I**	**M**

dove:

NU = Non usato.

T1 T0 = Determinano la durata del periodo di interrupt

0	0	->	1/64 secondo
0	1	->	1 secondo
1	0	->	1 minuto
1	1	->	1 ora

I = Determina modalità di gestione interrupt:

1 -> seleziona l'interrupt mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva con un reset del bit IF del registro D;

0 -> seleziona lo standard mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva autonomamente dopo 7,8 msec.

M = Maschera la generazione d'interrupt:

1 -> interrupt mascherato, il pin di interrupt é sempre disattivo.

0 -> interrupt non mascherato, il pin di interrupt riflette lo stato d'interrupt.

	D7	D6	D5	D4	D3	D2	D1	D0
REGF =	**NU**	**NU**	**NU**	**NU**	**T**	**24/12**	**S**	**R**

dove:

NU = Non usato.

T = Stabilisce da quale contatore interno prelevare il segnale di conteggio:

1 -> contatore principale (conteggio veloce per test);

0 -> 15° contatore (conteggio normale).

24/12 = Stabilisce il modo di conteggio delle ore:

1 -> 0÷23;

0 -> 0÷11 con AM/PM.

S = Se settato (1) provoca l'arresto dell'avanzamento dell'orologio fino alla sucessiva abilitazione (0).

R = Se settato (1) provoca il reset di tutti i contatori interni.

I registri del RTC utilizzano lo stesso indirizzo del registro RUNDEB, quindi ogni accesso ai registri del RTC implica anche l'acquisizione dell'ingresso di configurazione J1.
A seguito di un reset o power on nessuno dei registri del RTC é settato, mantenendo quindi i precedenti valori; se la circuiteria di back up é collegata i registri dati del RTC sono aggiornati anche quando la scheda non é alimentata.

PERIFERICHE DELLA CPU

La descrizione dei registri e del relativo significato di tutte le periferiche interne della CPU (Timer Counter, Port I/O, ADC, EEPROM, UART, ecc) é disponibile nell'appendice B. Qualora queste informazioni fossero ancora insufficienti, fare riferimento alla documentazione tecnica della casa costruttrice.

SCHEDE ESTERNE

La scheda **GPC® AM4** si interfaccia a buona parte dei moduli della serie BLOCK e di interfaccia operatore della **grifo®** od a molti altri sistemi di altre ditte. Le risorse di bordo possono essere facilmente aumentate collegandola alle numerose schede periferiche del carteggio **grifo®** tramite i suoi connettori standard. Anche schede in formato Europa con BUS **ABACO®** possono essere collegate, sfruttando gli appositi mother boards. A titolo di esempio ne riportiamo un elenco con una breve descrizione delle carratteristiche di massima; per maggiori informazioni richiedere o cercare l'apposita documentazione tecnica, sul CD o sul sito **grifo®**.

QTP G28

Quick Terminal Panel 28 tasti con LCD grafico

Interfaccia operatore provvista di display LCD grafico da 240x128 pixel retroilluminato con lampada a catodo freddo; tastiera a membrana da 28 tasti di cui 6 configurabili dall'utente; 16 LEDs di stato con attributo di lampeggio; alimentatore a bordo scheda; interdaccia seriale in RS 232, RS 422-485 o current loop; linea seriale ausiliaria in RS 232; controllore per linea CAN. Tasti ed etichette personalizzabili dall'utente tramite serigrafie da inserire in apposite tasche; contenitore metallico e plastico; EEPROM di set up; 256K EPROM o FLASH; Real Time Clock; 128K RAM; buzzer. Firmware di gestione che svolge funzione di terminale con primitive grafiche.

ADC 812

Analog to Digital Converter, 8 channels, 12 bits multi-range

Modulo periferico della serie 4 (100x50 mm); A/D converter DAS (Data Acquisition System) multi-range a 8 canali da 12 bit; Track-Hold; tempo di conversione 9µs; range dei segnali d'ingresso ±10, ±5, +10, +5Vdc oppure 0÷20, 4÷20mA; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

DAC 212

Digital to Analog Converter 12 bits, multi-range

Modulo periferico della serie 4 (100x50 mm); D/A converter multi-range a 2 canali da 12 bit; range del segnali d'uscita ± 10 o 0/+10 Vdc; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

CAN 14

Control Area Network, 1 channel, galvanically insulated

Modulo periferico della serie 4 (100x50 mm); UART CAN SJA1000; 1 canale seriale galvanicamente isolato; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

IBC 01

Interface Block Comunication

Scheda di conversioni per comunicazioni seriali. 2 linee RS 232; 1 linea RS 422-485; 1 linea in fibra ottica; interfaccia DTE/DCE selezionabile; attacco rapido per guide tipo DIN 46277-1 e 3.

KDL X24 - KDF 224

Keyboard Display LCD 2,4 righe 24 tasti - Keyboard Display Fluorescent 2 righe 24 tasti

Interfaccia tra 16 I/O TTL su connettore normalizzato I/O **ABACO®** a 20 vie e tastiera a matrice esterna da 24 tasti; display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs. Predisposizione per collegamento a tastiera telefonica.

QTP 24P

Quick Terminal Panel 24 tasti con interfaccia Parallela

Interfaccia operatore provvista di display alfanumerico fluorescente 20x 2 o LCD 20x2, 20x4 retroilluminato a LEDs; tastiera a membrana da 24 tasti di cui 12 configurabili dall'utente; 16 LEDs di stato; alimentatore a bordo scheda in grado di pilotare anche carichi esterni; interdaccia parallela basata su 16 I/O TTL di un connettore normalizzato I/O **ABACO®** a 20 vie. Tasti ed etichette personalizzabili tramite serigrafie da inserire in apposite tasche; opzione di contenitore metallico.

MCI 64

Memory Cards Interfaces 64 MBytes

Interfaccia per la gestione di Memory cards PCMCIA a 68 pins tramite un connettore normalizzato I/O **ABACO®**; sono disponibili driver per linguaggi ad alto livello.

ZBR xxx

Zipped BLOCK Relays xx Input + xx Output

Periferica per xx Input optoisolati e visualizzati tipo NPN; xx relé da 3A con MOV; connettori a morsettiera per ingressi optoisolati e uscite; connettore normalizzato **ABACO®** I/O BUS; LEDs di visualizzazione; sezione alimentatrice a bordo; attacco rapide per guide Ω. Le possibili configurazioni in termini di numero di I/O sono: xxx=324 con 32 In e 24 Out; xxx=246 con 24 In e 16 Out; xxx=168 con 16 In e 8 Out; xxx=84 con 8 In e 4 Out.

FBC xxx

Flat Block Contact xxx vie

Interfaccia tra 1 connettore a perforazione di isolante (scatolino da xx vie maschio) e la filatura da campo (morsettiere a rapida estrazione). Attacco rapido per guide tipo DIN 46277-1 e 3.

OBI N8 - OBI P8

Opto BLOCK Input NPN-PNP

Interfaccia per 8 input optoisolati e visualizzati tipo NPN, PNP, connettore a morsettiera, connettore standard I/O **ABACO®** a 20 vie; sezione alimentatrice; attacco rapido per guide DIN 46277-1 e 3.

TBO 01 - TBO 08

Transistor BLOCK Output

Interfaccia per 16 connettore normalizzato I/O **ABACO®** a 20 vie; 16 o 8 output a transistor in Open Collector da 45 Vcc 3 A su connettore a morsettiera. Uscite optoisolate e visualizzate; attacco rapido per guide DIN 6277-1 e 3.

ABB 03

Abaco® Block BUS 3 slots

Mother board **ABACO®** da 3 slots; passo 4 TE; guidaschede; connettori normalizzati di alimentazione; tasto di reset; LEDs per alimentazioni; interfaccia **ABACO®** I/O BUS. Attacco rapido per guide Ω.

IPC 52

Intelligent Peripheral Controller

Scheda periferica intelligente in grado di acquisire 24 segnali analogici generati da trasduttori da campo; 8 ingressi per PT 100, PT 1000; 8 ingressi per termocoppie J,K,S,T; 8 ingressi per segnali in tensione ±2 V o corrente 0÷20 mA; interrogazione tramite BUS **ABACO®** o tramite linea seriale in RS 232, RS 422-485 o current loop; 16 linee di I/O TTL; risoluzione di 16 bit più segno; 0,1 °C di precisione; 5 acquisizioni al secondo; funzionamento come data logher.

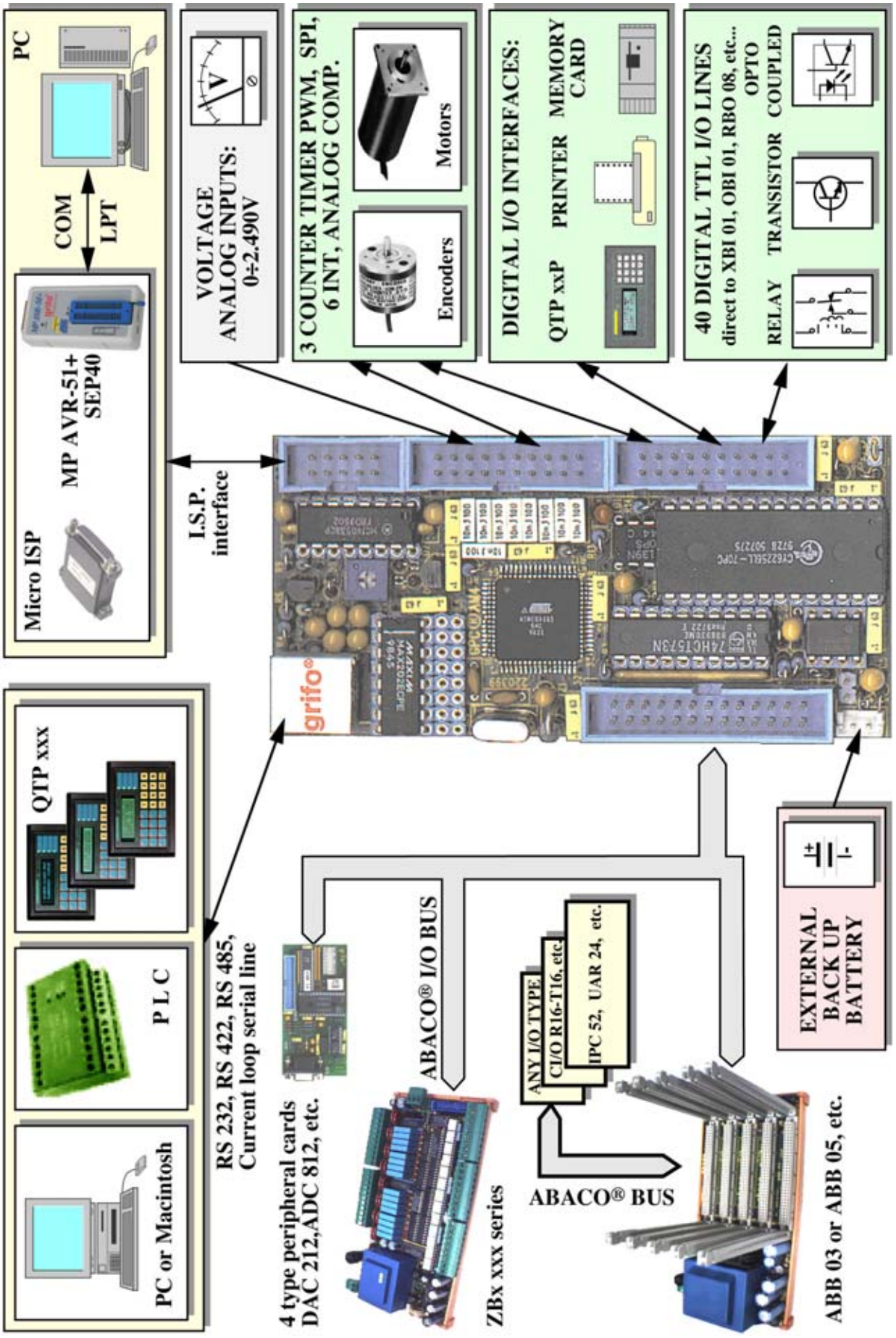


FIGURA 36: SCHEMA DELLE POSSIBILI CONNESSIONI

BIBLIOGRAFIA

E' riportato di seguito, un elenco di manuali e note tecniche, a cui l'utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti montati a bordo della scheda **GPC® AM4**.

Manuale ATMEL:	*AVR enhanced RISC microcontroller data book*
Manuale HEWLETT PACKARD:	*Optoelectronics Designer's Catalog*
Manuale MAXIM:	*New Releases Data Book - Volume IV*
Manuale NATIONAL SEMICONDUCTOR:	*Linear Data Book - Volume 1*
Manuale NEC:	*Memory Products*
Documentazione SEIKO EPSON:	*RTC 62421 Real Time Clock module*
Manuale TEXAS INSTRUMENTS:	*The TTL Data Book - SN54/74 Families*
Manuale TEXAS INSTRUMENTS:	*High Speed CMOS Logic Data Book*
Manuale TEXAS INSTRUMENTS:	*RS-422 and RS-485 Interface Circuits*
Manuale TEXAS INSTRUMENTS:	*Linear Circuits Voltage Regulators and Supervisor*

Per reperire questi manuali fare riferimento alle case produttrici ed ai relativi distributori locali. In alternativa si possono ricercare le medesime informazioni o gli eventuali aggiornamenti ai siti internet delle case elencate.

APPENDICE A: MONTAGGIO MECCANICO

La **GPC® AM4** può essere interfacciata al mondo esterno in due modalità; il primo é il cosidetto montaggio in piggy back, che consiste nel montare la scheda al di sopra del proprio hardware, sfruttando il prolungamento dei pin dei connettori CN1, CN5 e CN6. Questi infatti si estendono nel lato saldature per circa 7 mm, permettendo quindi un comodo inserimento su connettori femmina, del tipo strip a passo 2.54 mm.
La seconda modalità di montaggio, invece, consiste nell'inserire la scheda su una guida Weidmuller tipo RS/100 (codice 414487), per il montaggio su barre Ω del tipo DIN 46277-1 e 3; questo contenitore plastico può essere ordinato alla **grifo®** come opzione **WM4**. Con questo montaggio la scheda può essere anche abbinata ad altre schede periferiche (ad esempio un modulo tipo **ZBR** o **ZBT)**, ottenendo quindi un singolo elemento; in questo caso deve essere ordinata l'estensione della guida con il codice **.EXT-WMlll**, dove lll indica la lunghezza desiderata in mm.
In questo caso il collegamento elettrico tra la **GPC® AM4** e la scheda periferica avviene tramite un flat cable a 26 vie, che deve essere il più corto possibile, ed eventualmente può essere ordinato alla **grifo®**, con il codice **FLT 26+26 I/O**.
Nelle figure seguenti sono riportate le quote meccaniche, relative alla posizioni dei connettori ed alcune immagini riguardanti queste due modalità di connessione.

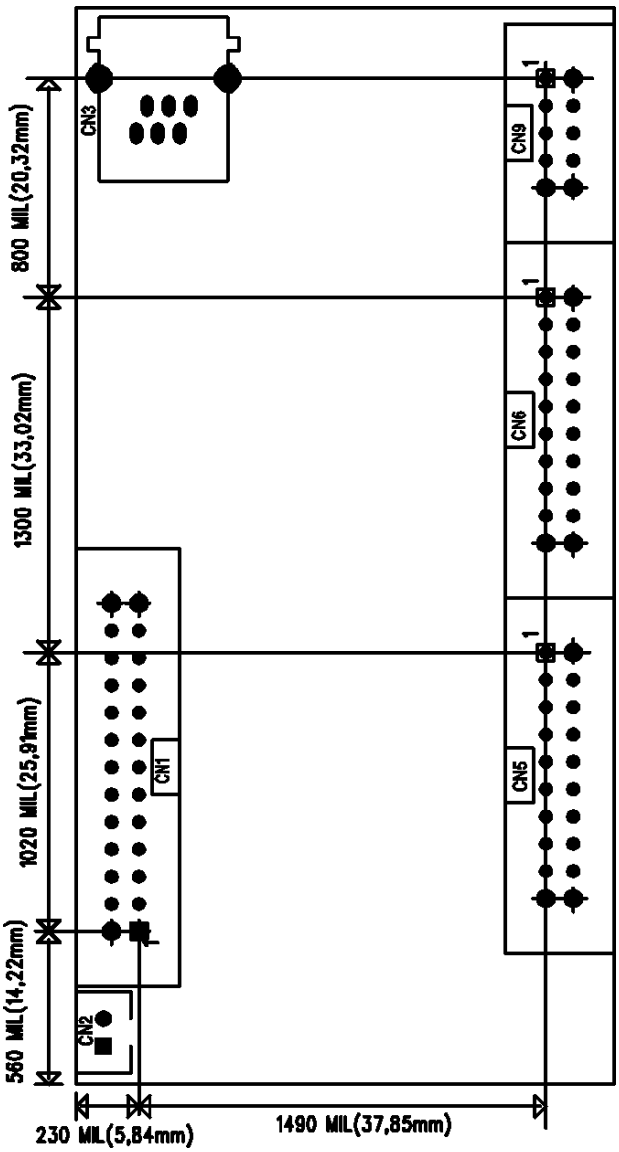


FIGURA A1: DIMA DI FORATURA PER MONTAGGIO IN PIGGY BACK

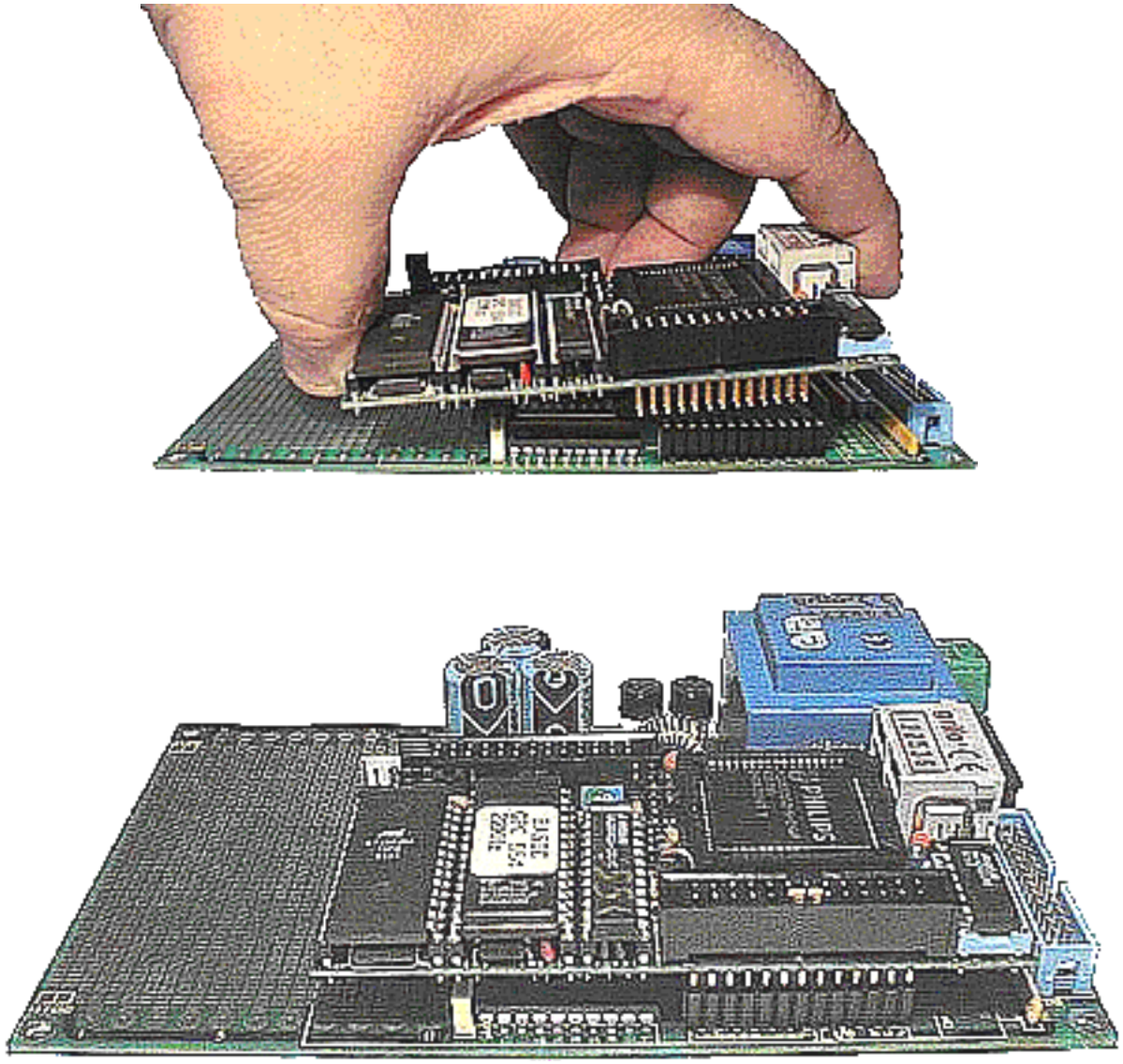


FIGURA A2: MONTAGGIO IN PIGGY BACK

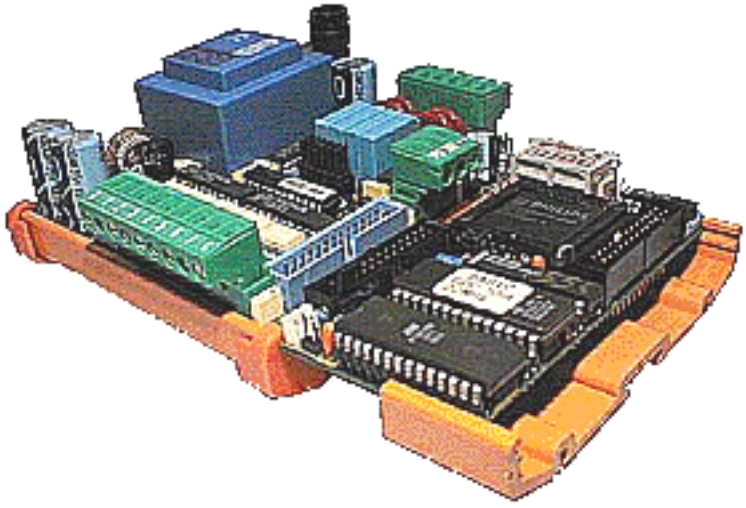


FIGURA A3: MONTAGGIO SU GUIDA WEIDMULLER

Features

- Utilizes the AVR® RISC Architecture
- AVR - High-performance and Low-power RISC Architecture
  - 120/121 Powerful Instructions - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Up to 6 MIPS Throughput at 6 MHz
- Data and Nonvolatile Program Memory
  - 64K/128K Bytes of In-System Programmable Flash
    Endurance: 1,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - 2K/4K Bytes of In-System Programmable EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program and EEPROM Data Security
  - SPI Interface for In-System Programming
- Peripheral Features
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - Programmable Serial UART
  - Master/Slave SPI Serial Interface
  - Real Time Counter (RTC) with Separate Oscillator
  - Two 8-bit Timer/Counters with Separate Prescaler and PWM
  - Expanded 16-bit Timer/Counter system, with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9-, or 10-bit PWM
  - Programmable Watchdog Timer with On-chip Oscillator
  - 8-channel, 10-bit ADC
- Special Microcontroller Features
  - Low-power Idle, Power Save and Power Down Modes
  - Software Selectable Clock Frequency
  - External and Internal Interrupt Sources
- Specifications
  - Low-power, High-speed CMOS Process Technology
  - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
  - Active: 5.5 mA
  - Idle Mode: 1.6 mA
  - Power Down Mode: < 1 µA
- I/O and Packages
  - 32 Programmable I/O Lines, 8 Output Lines, 8 Input Lines
  - 64-pin TQFP
- Operating Voltages
  - 2.7 - 3.6V (ATmega603L and ATmega103L)
  - 4.0 - 5.5V (ATmega603 and ATmega103)
- Speed Grades
  - 0 - 4 MHz (ATmega603L and ATmega103L)
  - 0 - 6 MHz (ATmega603 and ATmega103)

8-bit AVR® Microcontroller with 64K/128K Bytes In-Systen Programmable Flash

ATmega603
ATmega603L
ATmega103
ATmega103L

Preliminary

Rev. 0945C–04.

Features

- Utilizes the AVR® RISC Architecture
- AVR - High-performance and Low-power RISC Architecture
  - 120/121 Powerful Instructions - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Up to 6 MIPS Throughput at 6 MHz
- Data and Nonvolatile Program Memory
  - 64K/128K Bytes of In-System Programmable Flash
    Endurance: 1,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - 2K/4K Bytes of In-System Programmable EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program and EEPROM Data Security
  - SPI Interface for In-System Programming
- Peripheral Features
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - Programmable Serial UART
  - Master/Slave SPI Serial Interface
  - Real Time Counter (RTC) with Separate Oscillator
  - Two 8-bit Timer/Counters with Separate Prescaler and PWM
  - Expanded 16-bit Timer/Counter system, with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9-, or 10-bit PWM
  - Programmable Watchdog Timer with On-chip Oscillator
  - 8-channel, 10-bit ADC
- Special Microcontroller Features
  - Low-power Idle, Power Save and Power Down Modes
  - Software Selectable Clock Frequency
  - External and Internal Interrupt Sources
- Specifications
  - Low-power, High-speed CMOS Process Technology
  - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
  - Active: 5.5 mA
  - Idle Mode: 1.6 mA
  - Power Down Mode: < 1 µA
- I/O and Packages
  - 32 Programmable I/O Lines, 8 Output Lines, 8 Input Lines
  - 64-pin TQFP
- Operating Voltages
  - 2.7 - 3.6V (ATmega603L and ATmega103L)
  - 4.0 - 5.5V (ATmega603 and ATmega103)
- Speed Grades
  - 0 - 4 MHz (ATmega603L and ATmega103L)
  - 0 - 6 MHz (ATmega603 and ATmega103)

8-bit AVR® Microcontroller with 64K/128K Bytes In-Systen Programmable Flash

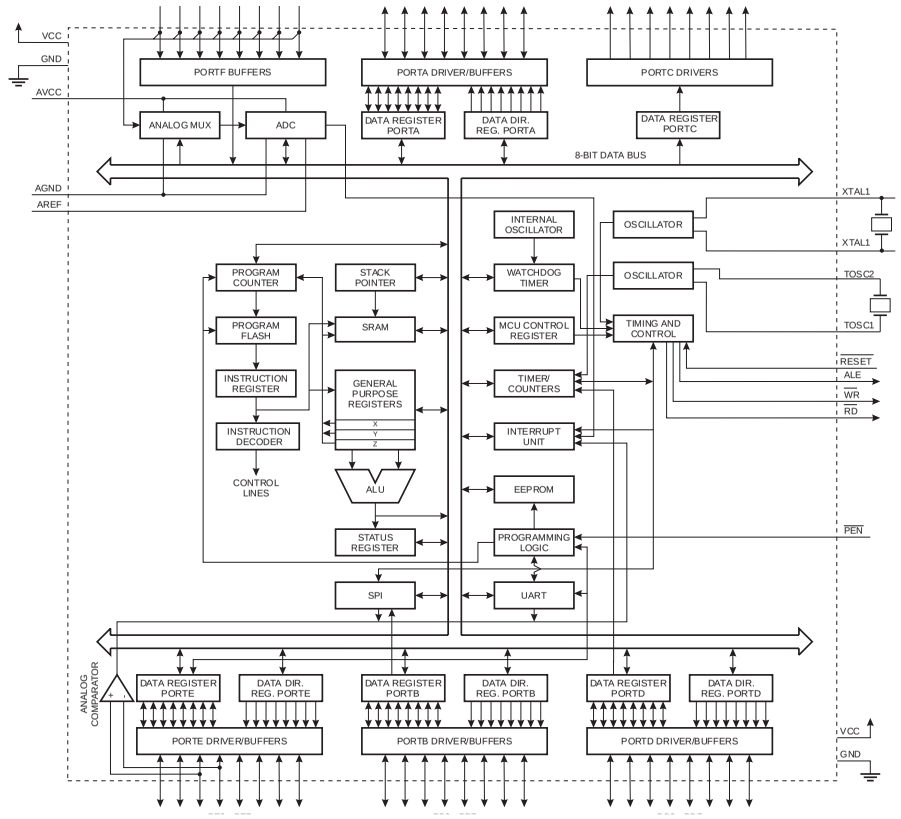
ATmega603
ATmega603L
ATmega103
ATmega103L

Preliminary

Rev. 0945C–04.

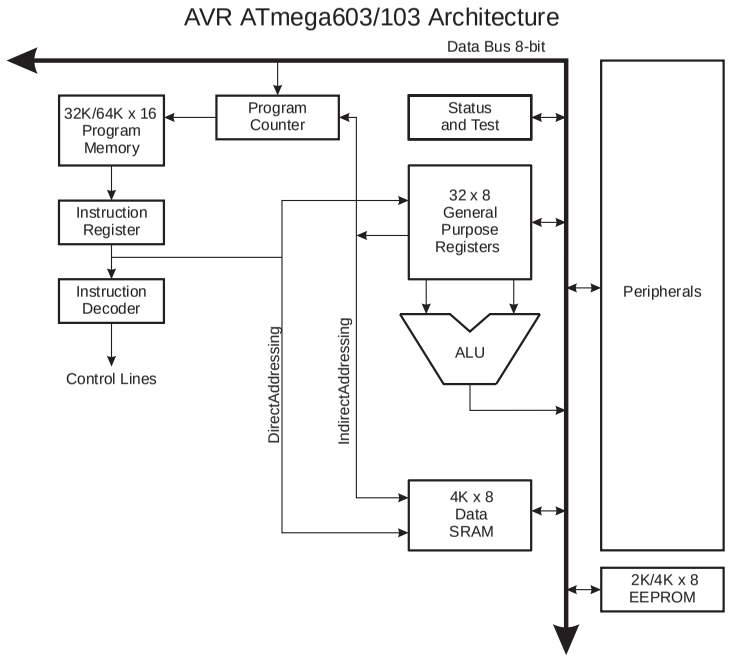
Block Diagram

Figure 1. The ATmega603/103 Block Diagram



Architectural Overview

Figure 4. The ATmega603/103 AVR RISC Architecture



The AVR uses a Harvard architecture concept - with separate memories and buses for program and data. The progra memory is accesses with a single level pipeline. While one instruction is being executed, the next instruction is pre-fetch from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory in-system programmable Flash memory. With a few exceptions, AVR instructions have a single 16-bit word forma meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack effectively allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size a the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts a executed). The 16-bit stack pointer SP is read/write accessible in the I/O space.

The 4000 bytes data SRAM can be easily accessed through the five different addressing modes supported in the AV architecture.

ITALIAN TECHNOLOGY

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the stati register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of tl program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower tl interrupt vector address, the higher the priority.

The memory spaces in the AVR architecture are all linear and regular memory maps.

General Purpose Register File

Figure 5 shows the structure of the 32 general purpose working registers in the CPU.

Figure 5. AVR CPU General Purpose Working Registers

	7 0	Addr.	
	R0	$00	
	R1	$01	
	R2	$02	
	. . .		
	R13	$0D	
General	R14	$0E	
Purpose	R15	$0F	
Working	R16	$10	
Registers	R17	$11	
	. . .		
	R26	$1A	X-register low byte
	R27	$1B	X-register high byte
	R28	$1C	Y-register low byte
	R29	$1D	Y-register high byte
	R30	$1E	Z-register low byte
	R31	$1F	Z-register high byte

All the register operating instructions in the instruction set have direct and single cycle access to all registers. The on exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and register and the LDI instruction for load immediate constant data. These instructions apply to the second half of the regi ters in the register file - R16..R31. The general SBC, SUB, CP, AND and OR and all other operations between two registe or on a single register apply to the entire register file.

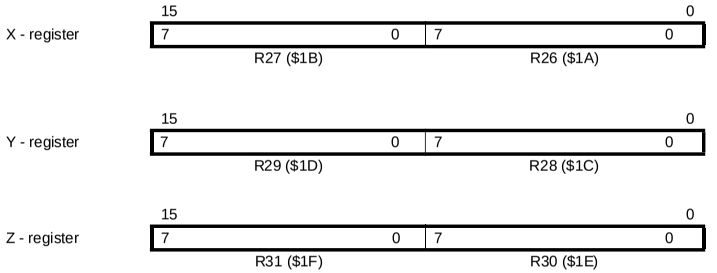
As shown in Figure 5, each register is also assigned a data memory address, mapping them directly into the first 3 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organizatic provides great flexibility in access of the registers, as the X,Y and Z registers can be set to index any register in the file.

The 4K bytes of SRAM available for general data are implemented as addresses $0060 to $0FFF.

X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are address pointe for indirect addressing of the SRAM. The three indirect address registers X, Y and Z are defined as:

Figure 6. X, Y and Z Registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment ar decrement (see the descriptions for the different instructions).

ALU - Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided in three main categories - arithmetic, logical and bit-functions.

ISP Flash Program Memory

The ATmega603/103 contains 64K/128K bytes on-chip In-system Programmable Flash memory for program storag Since all instructions are single or double 16-bit words, the Flash is organized as 64K x 16. The Flash memory has a endurance of at least 1000 write/erase cycles.

Constant tables can be allocated in the entire program memory space (see the LPM - Load Program Memory and ELP Extended Load Program Memory instruction descriptions).

SRAM Data Memory

The ATmega603/103 supports two different configurations for the SRAM data memory as listed in the following table:

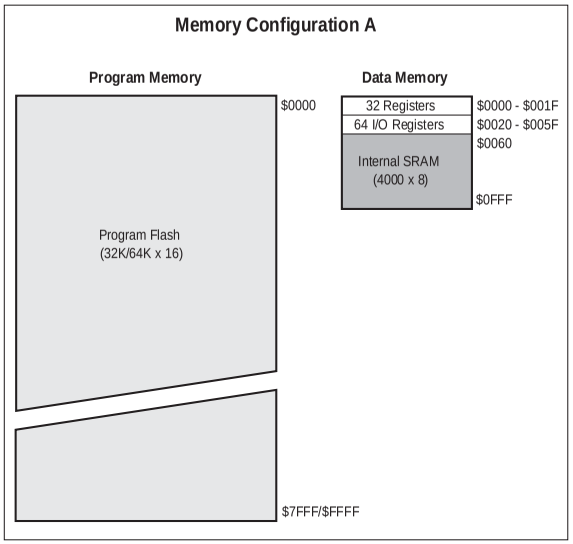
Table 2. Memory Configurations

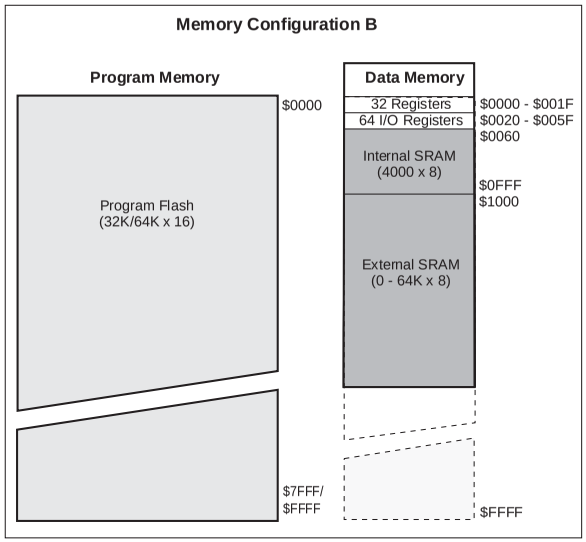
Configuration	Internal SRAM Data Memory	External SRAM Data Memory
A	4000	None
B	4000	up to 64K

Note: When using 64K of External SRAM, 60K will be available.

ITALIAN TECHNOLOGY grifo® abaco bus

Figure 7. Memory Configurations





The 4096 first Data Memory locations address both the Register file, the I/O Memory and the internal data SRAM. The fi 96 locations address the register file and I/O memory, and the next 4000 locations address the internal data SRAM.

An optional external data SRAM can be used with the ATmega603/103. This SRAM will occupy an area in the remaini address locations in the 64K address space. This area starts at the address following the internal SRAM. If a 64K exter SRAM is used, 4K of the external memory is lost as the addresses are occupied by internal memory.

When the addresses accessing the SRAM memory space exceeds the internal data memory locations, the external da SRAM is accessed using the same instructions as for the internal data memory access. When the internal data memori are accessed, the read and write strobe pins (RD and WR) are inactive during the whole access cycle. External SRA operation is enabled by setting the SRE bit in the MCUCR register.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This mea that the commands LD, ST, LDS, STS, PUSH and POP take one additional clock cycle. If the stack is placed in exter SRAM, interrupts, subroutine calls and returns take two clock cycles extra because the two-byte program counter pushed and popped. When external SRAM interface is used with wait state, two additional clock cycles is used per byt This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls a returns will need four clock cycles more than specified in the instruction set manual.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect wi Pre-Decrement and Indirect with Post-Increment. In the register file, registers R26 to R31 feature the indirect addressi pointer registers.

The Indirect with Displacement mode features a 63 address locations reach from the base address given by the Y Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers Y and Z are decremented and incremented.

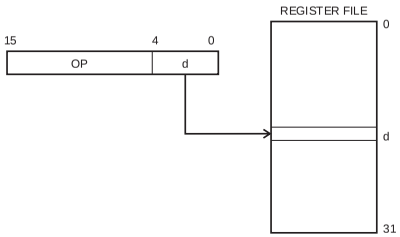
The entire data address space including the 32 general purpose working registers and the 64 I/O registers are all acces ble through all these addressing modes. See the next section for a detailed description of the different addressing mode

Program and Data Addressing Modes

The ATmega603/103 AVR RISC microcontroller supports powerful and efficient addressing modes for access to t program memory (Flash) and data memory (SRAM, Register File and I/O Memory). This section describes the differe addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instructi word. To simplify, not all figures show the exact location of the addressing bits.

Register Direct, Single Register Rd

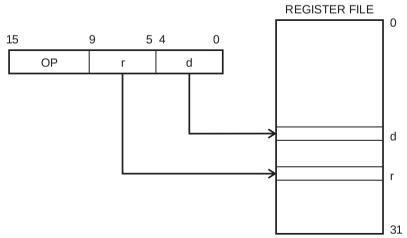
Figure 8. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Direct, Two Registers Rd and Rr

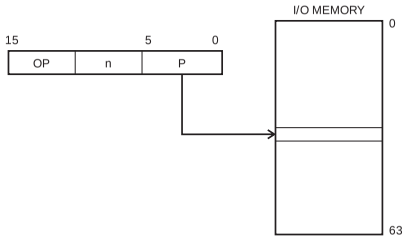
Figure 9. Direct Register Addressing, Two Registers



Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O Direct

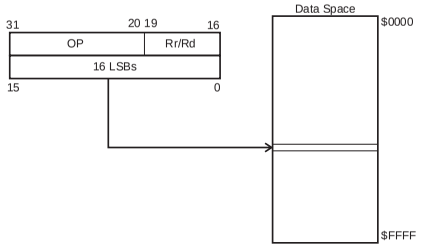
Figure 10. I/O Direct Addressing



Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

Data Direct

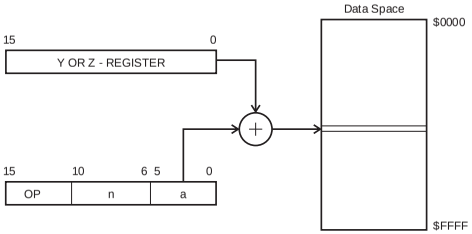
Figure 11. Direct Data Addressing



A 16-bit Data Address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or sour register.

Data Indirect with Displacement

Figure 12. Data Indirect with Displacement

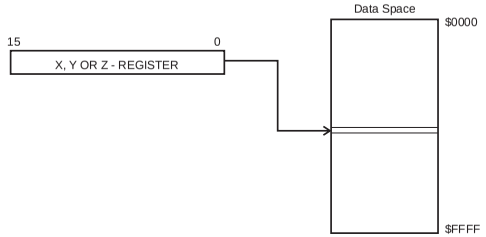


Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instructi word.

ITALIAN TECHNOLOGY grifo® abaco bus

Data Indirect

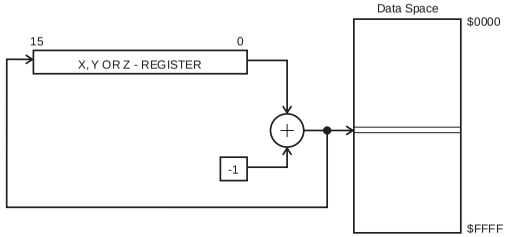
Figure 13. Data Indirect Addressing



Operand address is the contents of the X, Y or the Z-register.

Data Indirect With Pre-Decrement

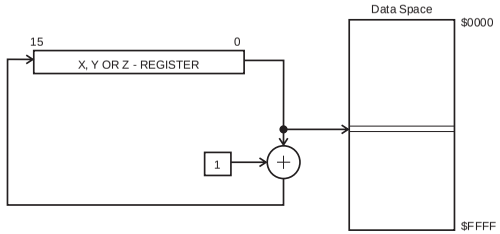
Figure 14. Data Indirect Addressing with Pre-Decrement



The X, Y or the Z-register is decremented before the operation. Operand address is the decremented contents of the X, or the Z-register.

Data Indirect With Post-Increment

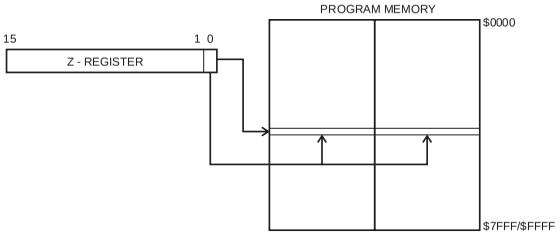
Figure 15. Data Indirect Addressing with Post-Increment



The X, Y or the Z-register is incremented after the operation. Operand address is the content of the X, Y or the Z-regist prior to incrementing.

Constant Addressing Using the LPM and ELPM Instructions

Figure 16. Code Memory Constant Addressing

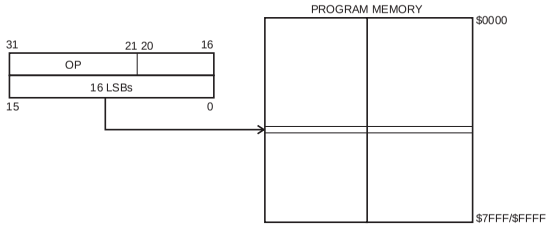


Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 32K), LSB selects l byte if cleared (LSB = 0) or high byte if set (LSB = 1). If ELPM is used, LSB of the RAM Page Z register - RAMPZ is used select low or high memory page (RAMPZ0 = 0: Low Page, RAMPZ0 = 1: High Page). ELPM does not apply to t ATmega603.

ITALIAN TECHNOLOGY

Direct Program Address, JMP and CALL

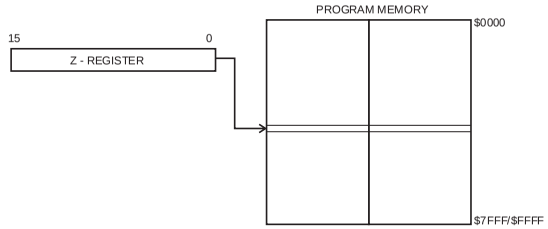
Figure 17. Direct Program Memory Addressing



Program execution continues at the address immediate in the instruction words.

Indirect Program Addressing, IJMP and ICALL

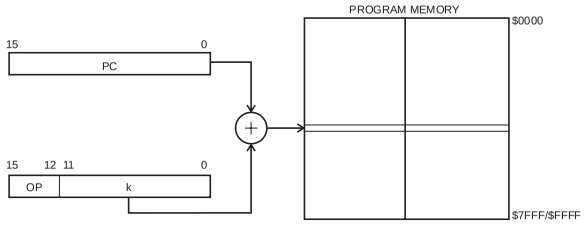
Figure 18. Indirect Program Memory Addressing



Program execution continues at address contained by the Z-register (i.e. the PC is loaded with the contents of tl Z-register).

Relative Program Addressing, RJMP and RCALL

Figure 19. Relative Program Memory Addressing



Program execution continues at address PC + k + 1. The relative address k is -2048 to 2047.

EEPROM Data Memory

The EEPROM memory is organized as a separate data space, in which single bytes can be read and written. TI EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU described on page 52 specifying the EEPROM address register, the EEPROM data register, and the EEPROM contr register.

Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock Ø, directly generated from the external clock crystal for the chip. No interr clock division is used.

Figure 20 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and tl fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the correspondi unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 20. The Parallel Instruction Fetches and Instruction Executions

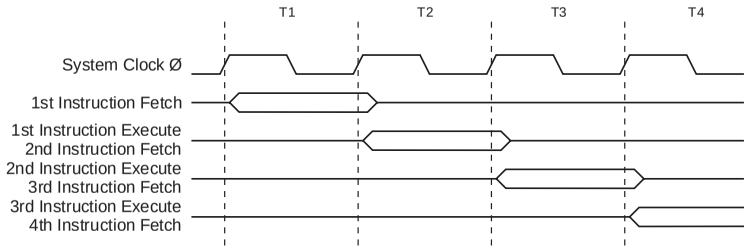


Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two regist operands is executed, and the result is stored back to the destination register.

GPC® AM4 Rel. 3.00
Pagina B-7
ITALIAN TECHNOLOGY
grifo®
abaco
bus

I/O Memory

The I/O space definition of the ATmega603/103 is shown in the following table:

Table 3. ATmega603/103 I/O Space

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status REGister
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3C ($5C)	XDIV	XTAL Divide Control Register
$3B ($5B)	RAMPZ	RAM Page Z Select Register
$3A ($5A)	EICR	External Interrupt Control Register
$39 ($59)	EIMSK	External Interrupt MaSK register
$38 ($58)	EIFR	External Interrupt Flag Register
$37 ($57)	TIMSK	Timer/Counter Interrupt MaSK register
$36 ($56)	TIFR	Timer/Counter Interrupt Flag register
$35 ($55)	MCUCR	MCU General Control Register
$34 ($54)	MCUSR	MCU Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register
$30 ($50)	ASSR	Asynchronous Mode Status Register
$2F ($4F)	TCCR1A	Timer/Counter1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter1 Low Byte
$2B ($4B)	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
$29 ($49)	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
$28 ($48)	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter1 Input Capture Register Low Byte
$25 ($45)	TCCR2	Timer/Counter2 Control Register
$24 ($44)	TCNT2	Timer/Counter2 (8-bit)
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1F ($3F)	EEARH	EEPROM Address Register High
$1E ($3E)	EEARL	EERPOM Address Register Low
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register

Table 3. ATmega603/103 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$1B ($3B)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$15 ($35)	PORTC	Data Register, Port C
$12 ($32)	PORTD	Data Register, Port D
$11 ($31)	DDRD	Data Direction Register, Port D
$10 ($30)	PIND	Input Pins, Port D
$0F ($2F)	SPDR	SPI I/O Data Register
$0E ($2E)	SPSR	SPI Status Register
$0D ($2D)	SPCR	SPI Control Register
$0C ($2C)	UDR	UART I/O Data Register
$0B ($2B)	USR	UART Status Register
$0A ($2A)	UCR	UART Control Register
$09 ($29)	UBRR	UART Baud Rate Register
$08 ($28)	ACSR	Analog Comparator Control and Status Register
$07 ($27)	ADMUX	ADC Multiplexer Select Register
$06 ($26)	ADCSR	ADC Control and Status Register
$05 ($25)	ADCH	ADC Data Register High
$04 ($24)	ADCL	ADC Data Register Low
$03 ($23)	PORTE	Data Register, Port E
$02 ($22)	DDRE	Data Direction Register, Port E
$01 ($21)	PINE	Input Pins, Port E
$00 ($20)	PINF	Input Pins, Port F

Note: Reserved and unused locations are not shown in the table

All the different ATmega603/103 I/Os and peripherals are placed in the I/O space. The different I/O locations are direc accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I. space. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instructi set chapter for more details. When using the I/O specific instructions IN, OUT, the I/O register address $00 - $3F are use When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout th document are shown with the SRAM address in parentheses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory address should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructio work with registers $00 to $1F only.

The different I/O and peripherals control registers are explained in the following sections.

Status Register - SREG

The AVR status register - SREG - at I/O space location $3F ($5F) is defined as:

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - I: Global Interrupt Enable**

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control then performed in separate control registers. If the global interrupt enable register is cleared (zero), none of the interrup are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt h occurred, and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 - T: Bit Copy Storage**

The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as source and destination for the operated bit. bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in register in the register file by the BLD instruction.

- **Bit 5 - H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detail information.

- **Bit 4 - S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instru tion Set Description for detailed information.

- **Bit 3 - V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description f detailed information.

- **Bit 2 - N: Negative Flag**

The negative flag N indicates a negative result from an arithmetical or logical operation. See the Instruction Set Descripti for detailed information.

- **Bit 1 - Z: Zero Flag**

The zero flag Z indicates a zero result from an arithmetical or logical operation. See the Instruction Set Description f detailed information.

- **Bit 0 - C: Carry Flag**

The carry flag C indicates a carry in an arithmetical or logical operation. See the Instruction Set Description for detail information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning fro an interrupt routine. This must be handled by software.

Stack Pointer - SP

The general AVR 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E ($5E) a $3D ($5D). As the ATmega603/103 supports up to 64 kB memory, all 16-bits are used.

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
$3D ($5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Sta space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts a enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is push onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack w subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the PO instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET return from interrupt RETI.

RAM Page Z Select Register - RAMPZ

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	-	-	-	-	-	-	-	RAMPZ0	RAMPZ
Read/Write	R	R	R	R	R	R	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

The RAMPZ register is normally used to select which 64K RAM Page is accessed by the Z pointer. As the ATmega603/1 does not support more than 64K of SRAM memory, this register is used only to select which page in the program memo is accessed when the ELPM instruction is used. The different settings of the RAMPZ0 bit have the following effects:

RAMPZ0 = 0: Program memory address $0000- $7FFF (lower 64K bytes) is accessed by ELPM

RAMPZ0 = 1: Program memory address $8000- $FFFF (higher 64K bytes) is accessed by ELPM

Note that LPM is not affected by the RAMPZ setting.

The ATmega603 does not contain the RAMPZ register, and it does not have the ELPM instruction. The ordinary LP instruction can reach the entire program memory in the ATmega603.

MCU Control Register - MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	SRE	SRW	SE	SM1	SM0	-	-	-	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - SRE: External SRAM Enable**

When the SRE bit is set (one), the external data SRAM is enabled, and the pin functions AD0-7 (Port A), and A8-15 (P C) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective da direction registers. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin and da direction settings are used.

- **Bit 6 - SRW: External SRAM Wait State**

When the SRW bit is set (one), a one cycle wait state is inserted in the external data SRAM access cycle. When the SR bit is cleared (zero), the external data SRAM access is executed with a three-cycle scheme. See Figure 51.External SRAM Access Cycle without wait states73 and Figure 52.External SRAM Access Cycle with wait state74.

- **Bit 5 - SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid t MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit ju before the execution of the SLEEP instruction.

• **Bits 4,3 - SM1/SM0: Sleep Mode Select bits 1 and 0**

This bit selects between the three available sleep modes as shown in the following table:

Table 4. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle Mode
0	1	Reserved
1	0	Power Down
1	1	Power Save

• **Bits 2..0 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and always read zero.

XTAL Divide Control Register - XDIV

The XTAL Divide Control Register is used to divide the XTAL clock frequency by a number in the range 1 - 129. This fe ture can be used to decrease power consumption when the requirement for processing power is low.

Bit	7	6	5	4	3	2	1	0	
$3C ($5C)	**XDIVEN**	**XDIV6**	**XDIV5**	**XDIV4**	**XDIV3**	**XDIV2**	**XDIV1**	**XDIV0**	**XDIV**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

• **Bit 7 - XDIVEN: XTAL Divide Enable**

When the XDIVEN bit is set (one), the clock frequency of the CPU and all peripherals is divided by the factor defined by t setting of XDIV6 - XDIV0. This bit can be set and cleared run-time to vary the clock frequency as suitable to the applicatic

• **Bits 6..0 - XDIV6..XDIV0: XTAL Divide Select Bits 6 - 0**

These bits define the division factor that applies when the XDIVEN bit is set (one). If the value of these bits is denoted the following formula defines the resulting CPU clock frequency f_{clk}:

$$f_{CLK} = \frac{XTAL}{129 - d}$$

The value of these bits can only be changed when XDIVEN is zero. When XDIVEN is set to one, the value written simult neously into XDIV6..XDIV0 is taken as the division factor. When XDIVEN is cleared to zero, the value writt simultaneously into XDIV6..XDIV0 is rejected. As the divider divides the master clock input to the MCU, the speed of peripherals is reduced when a division factor is used.

Reset and Interrupt Handling

The ATmega603/103 provides 23 different interrupt sources. These interrupts and the separate reset vector each have separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be s (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. Tl complete list of vectors is shown in Table 5. The list also determines the priority levels of the different interrupts. The low the address the higher is the priority level. RESET has the highest priority, and next is INT0 - the External Interrupt Reque 0 etc.

Table 5. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$0000	RESET	Hardware Pin, Power-on Reset and Watchdog Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	$0022	SPI, STC	SPI Serial Transfer Complete
19	$0024	UART, RX	UART, Rx Complete
20	$0026	UART, UDRE	UART Data Register Empty
21	$0028	UART, TX	UART, Tx Complete
22	$002A	ADC	ADC Conversion Complete
23	$002C	EE READY	EEPROM Ready
24	$002E	ANALOG COMP	Analog Comparator

The most typical program setup for the Reset and Interrupt Vector Addresses are:

```
Address   Labels     Code                 Comments
$0000               jmp    RESET         ; Reset Handler
$0002               jmp    EXT_INT0      ; IRQ0 Handler
$0004               jmp    EXT_INT1      ; IRQ1 Handler
$0006               jmp    EXT_INT2      ; IRQ2 Handler
$0008               jmp    EXT_INT3      ; IRQ3 Handler
$000A               jmp    EXT_INT4      ; IRQ4 Handler
$000C               jmp    EXT_INT5      ; IRQ5 Handler
$000E               jmp    EXT_INT6      ; IRQ6 Handler
$0010               jmp    EXT_INT7      ; IRQ7 Handler
$0012               jmp    TIM2_COMP     ; Timer2 Compare Handler
$0014               jmp    TIM2_OVF      ; Timer2 Overflow Handler
$0016               jmp    TIM1_CAPT     ; Timer1 Capture Handler
$0018               jmp    TIM1_COMPA    ; Timer1 CompareA Handler
$001A               jmp    TIM1_COMPB    ; Timer1 CompareB Handler
$001C               jmp    TIM1_OVF      ; Timer1 Overflow Handler
$001E               jmp    TIM0_COMP     ; Timer0 Compare Handler
$0020               jmp    TIM0_OVF      ; Timer0 Overflow Handler
$0022               jmp    SPI_STC       ; SPI Transfer Complete Handler
$0024               jmp    UART_RXC      ; UART RX Complete Handler
$0026               jmp    UART_DRE      ; UDR Empty Handler
$0028               jmp    UART_TXC      ; UART TX Complete Handler
$002A               jmp    ADC           ; ADC Conversion Complete Handler
$002C               jmp    EE_RDY        ; EEPROM Ready Handler
$002E               jmp    ANA_COMP      ; Analog Comparator Handler
;
$0030     MAIN:     ldi    r16, high(RAMEND); Main program start
$0031               out    SPH,r16
$0032               ldi    r16, low(RAMEND)
$0033               out    SPL,r16
$0034               <instr>  xxx
  ...      ...        ...    ...
```

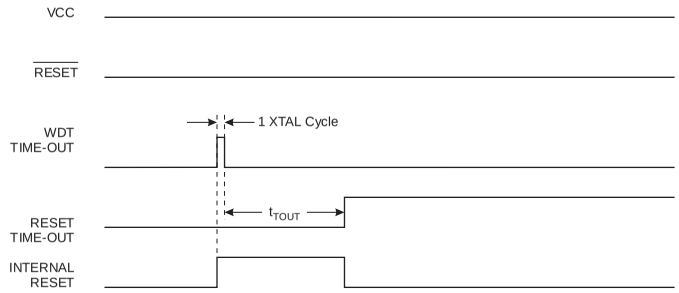
Reset Sources

The ATmega603/103 has three sources of reset:

- Power-On Reset. The MCU is reset when the supply voltage is below the power-on reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers except the MCU Status register are then set to their initial values, and the program starts ex cution from address $0000. The instruction placed in address $0000 must be a JMP - absolute jump instruction to the res handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular progra code can be placed at these locations. The circuit diagram in Figure 23 shows the reset logic. Table 6 defines the timi and electrical parameters of the reset circuitry.

Figure 27. Watchdog Reset During Operation



MCU Status Register - MCUSR

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
$34 ($54)	-	-	-	-	-	-	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	See bit description		

- **Bits 7..2 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always read as zero.

- **Bit 1 - EXTRF: External Reset Flag**

After a power-on reset, this bit is undefined (X). It will be set by an external reset. A watchdog reset will leave this unchanged.

- **Bit 0 - PORF: Power-on Reset Flag**

This bit is set by a power-on reset. A watchdog reset or an external reset will leave this bit unchanged.

To summarize, the following table shows the value of these two bits after the three modes of reset:

Table 8. PORF and EXTRF Values after Reset

Reset Source	EXTRF	PORF
Power-on Reset	undefined	1
External Reset	1	unchanged
Watchdog Reset	unchanged	unchanged

To make use of these bits to identify a reset condition, the user software should clear both the PORF and EXTRF bits early as possible in the program. Checking the PORF and EXTRF values is done before the bits are cleared. If the bit cleared before an external or watchdog reset occurs, the source of reset can be found by using the following truth table:

Table 9. Reset Source Identification

EXTRF	PORF	Reset Source
0	0	Watchdog Reset
0	1	Power-on Reset
1	0	External Reset
1	1	Power-on Reset

Interrupt Handling

The ATmega603/103 has two dedicated 8-bit Interrupt Mask control registers; EIMSK - External Interrupt Mask register a TIMSK - Timer/Counter Interrupt Mask register. In addition, other enable and mask bits can be found in the periphe control registers.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user so ware can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction - RE - is executed.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, ha ware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing logic one to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be s and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interru flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priori

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition active.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning fro an interrupt routine. This must be handled by software.

External Interrupt Mask Register - EIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - INT7 - INT4: External Interrupt Request 7-4 Enable**

When an INT7- INT4 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the corresponding external p interrupt is enabled. The Interrupt Sense Control bits in the External Interrupt Control Register - EICR defines whether t external interrupt is activated on rising or falling edge or level sensed. Activity on any of these pins will trigger an interru request even if the pin is enabled as an output. This provides a way of generating a software interrupt.

- **Bits 3..0 - INT3 - INT0: External Interrupt Request 3-0 Enable**

When an INT3 - INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the corresponding external p interrupt is enabled. The external interrupts are always low level triggered interrupts. Activity on any of these pins will tr ger an interrupt request even if the pin is enabled as an output. This provides a way of generating a software interru When enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low.

External Interrupt Flag Register - EIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	INTF7	INTF6	INTF5	INTF4	-	-	-	-	EIFR
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - INTF7 - INTF4: External Interrupt 7-4 Flags**

When an event on the INT7 - INT4 pins triggers an interrupt request, the corresponding interrupt flag, INTF7 - INT becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT7 - INT4 in EIMSK, are set (one), t MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag cleared by writing a logical one to it.

- **Bits 3..0 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always read as zero.

External Interrupt Control Register - EICR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - ISCX1, ISCX0: External Interrupt 7-4 Sense Control bits**

The External Interrupts 7 - 4 are activated by the external pins INT7 - INT4 if the SREG I-flag and the corresponding inte rupt mask in the EIMSK is set. The level and edges on the external pins that activate the interrupts are defined in t following table:

Table 10. Interrupt Sense Control

ISCX1	ISCX0	Description
0	0	The low level of INTX generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INTX generates an interrupt request.
1	1	The rising edge of INTX generates an interrupt request.

Note: X = 7, 6, 5 or 4.
When changing the ISCX1/ISCX0 bits, the interrupt must be disabled by clearing its Interrupt Enable bit in the GIMSK Registe Otherwise an interrupt can occur when the bits are changed.

The value on the INTX pin is sampled before detecting edges. If edge interrupt is selected, pulses that last longer than o CPU clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. Observe that CP clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. If low level interrupt is selected, t low level must be held until the completion of the currently executing instruction to generate an interrupt. If enabled, a lev triggered interrupt will generate an interrupt request as long as the pin is held low.

Timer/Counter Interrupt Mask Register - TIMSK

Bit	7	6	5	4	3	2	1	0	
$37 ($57)	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - OCIE2: Timer/Counter2 Output Compare Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match int rupt is enabled. The corresponding interrupt (at vector $0012) is executed if a Compare match in Timer/Counter2 occu i.e., when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 6 - TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt enabled. The corresponding interrupt (at vector $0014) is executed if an overflow in Timer/Counter2 occurs, i.e., when t TOV2 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 5 - TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Eve Interrupt is enabled. The corresponding interrupt (at vector $0016) is executed if a capture-triggering event occurs on p 29, PD4(IC1), i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 4 - OCE1A: Timer/Counter1 Output CompareA Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Mat interrupt is enabled. The corresponding interrupt (at vector $0018) is executed if a CompareA match in Timer/Counte occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

• **Bit 3 - OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Mat interrupt is enabled. The corresponding interrupt (at vector $001A) is executed if a CompareB match in Timer/Counte occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

• **Bit 2 - TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt enabled. The corresponding interrupt (at vector $001C) is executed if an overflow in Timer/Counter1 occurs, i.e., when tl TOV1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

• **Bit 1 - OCIE0: Timer/Counter0 Output Compare Interrupt Enable**

When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match inte rupt is enabled. The corresponding interrupt (at vector $001E) is executed if a Compare match in Timer/Counter0 occu i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

• **Bit 0 - TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt enabled. The corresponding interrupt (at vector $0020) is executed if an overflow in Timer/Counter0 occurs, i.e., when tl TOV0 bit is set in the Timer/Counter Interrupt Flag Register - TIFR

Timer/Counter Interrupt Flag Register - TIFR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

• **Bit 7 - OCF2: Output Compare Flag 2:**

The OCF2 bit is set (one) when compare match occurs between Timer/Counter2 and the data in OCR2 - Output Compa Register 2. OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCI is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare Interru Enable), and the OCF2 are set (one), the Timer/Counter2 Output Compare Interrupt is executed.

• **Bit 6 - TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing tl corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the I-bit SREG, and TOIE2 (Timer/Counter1 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflc Interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 advances from $00.

• **Bit 5 - ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to tl input capture register - ICR1. ICF1 is cleared by hardware when executing the corresponding interrupt handling vect Alternatively, ICF1 is cleared by writing a logic one to the flag. When the SREG I-bit, and TICIE1 (Timer/Counter1 Inp Capture Interrupt Enable), and ICF1 are set (one), the Timer/Counter1 Capture Interrupt is executed.

• **Bit 4 - OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A - Outp Compare Register 1A. OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Altern tively, OCF1A is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compa Interrupt Enable), and the OCF1A are set (one), the Timer/Counter1 Compare A match Interrupt is executed.

• **Bit 3 - OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B - Outp Compare Register 1B. OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Altern tively, OCF1B is cleared by writing a logic one to the flag.. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compa match Interrupt Enable), and the OCF1B are set (one), the Timer/Counter1 Compare B match Interrupt is executed.

• **Bit 2 - TOV1: Timer/Counter1 Overflow Flag**

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the cc responding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic one to the flag. When the I-bit SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflc Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 advances from $0000.

• **Bit 1 - OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when compare match occurs between Timer/Counter0 and the data in OCR0 - Output Compa Register 0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCI is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE0 (Timer/Counter2 Compare Interru Enable), and the OCF0 are set (one), the Timer/Counter0 Output Compare Interrupt is executed.

• **Bit 0 - TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing tl corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interru is executed. In PWM mode, this bit is set when Timer/Counter0 advances from $00.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after t interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by The vector is normally a jump to the interrupt routine, and this jump takes 3 clock cycles. If an interrupt occurs during ex cution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clo cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When tl AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pendi interrupt is served.

Sleep Modes

To enter any of the three sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be execute The SM1 and SM0 bits in the MCUCR register select which sleep mode (Idle, Power Down, or Power Save) will be ac vated by the SLEEP instruction, see Table 4.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine, a resumes execution from the instruction following SLEEP. The contents of the register file, SRAM, and I/O memory are un tered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector

Idle Mode

When the SM1/SM0 bits are set to 00, the SLEEP instruction makes the MCU enter the Idle Mode, stopping the CPU b allowing SPI, UART, Analog Comparator, ADC, Timer/Counters, Watchdog and the interrupt system to continue operatir This enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow a UART Receive Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the analog compa tor can be powered down by setting the ACD-bit in the Analog Comparator Control and Status register - ACSR. This v reduce power consumption in Idle Mode. When the MCU wakes up from Idle mode, the CPU starts program executi immediately.

Power Down Mode

When the SM1/SM0 bits are set to 10, the SLEEP instruction makes the MCU enter the Power Down Mode. In this mod the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only external reset, a watchdog reset (if enabled), or an external level interrupt can wake up the MCU.

Note that if a level triggered interrupt is used for wake-up from Power Down Mode, the changed level must be held some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by t watchdog oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of t watchdog oscillator is 1 us (nominal) at 5.0V and 25C. The frequency of the watchdog oscillator is voltage dependent shown in section “Typical characteristics” on page 110.

When waking up from Power Down Mode, there is a delay from the wake-up condition occurs until the wake-up becom effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined the same SUT fuses that define the reset time-out period. The wake-up period is equal to the clock reset period, as sho in Table 6.

If the wake-up condition disappears before the MCU wakes up and starts to execute, e.g. a low level on is not held lo enough, the interrupt causing the wake-up will not be executed.

Power Save Mode

When the SM1/SM0 bits are 11, the SLEEP instruction makes the MCU enter the Power Save Mode. This mode is identi to Power Down, with one exception:

If Timer/Counter0 is clocked asynchronously, i.e. the AS0 bit in ASSR is set, Timer/Counter0 will run during sleep. In ad tion to the Power Down wake-up sources, the device can also wake up from either Timer Overflow or Output Compa event from Timer/Counter0 if the corresponding Timer/Counter0 interrupt enable bits are set in TIMSK. To ensure that t part executes the Interrupt routine when waking up, also set the global interrupt enable bit in SREG.

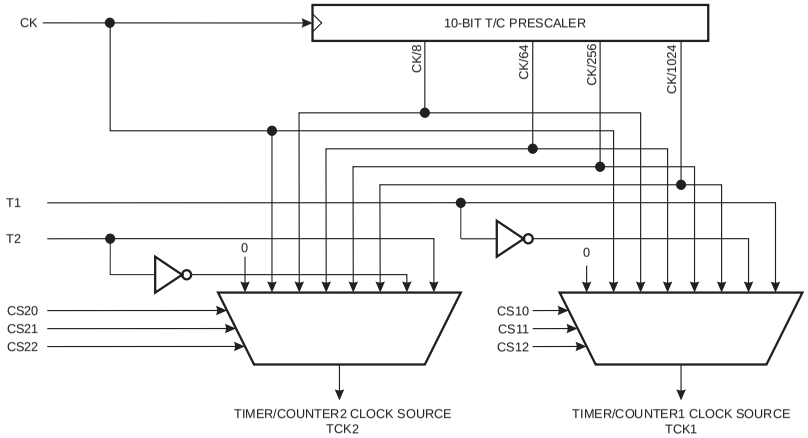
When waking up from Power Save Mode by an external interrupt, 2 instruction cycles are executed before the interru flags are updated. When waking up by the asynchronous timer, 3 instruction cycles are executed before the flags a updated. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and t interrupt routine has not started yet.

Timer/Counters

The ATmega603/103 provides three general purpose Timer/Counters - two 8-bit T/Cs and one 16-bit T/C. Timer/Counte can optionally be asynchronously clocked from an external oscillator. This oscillator is optimized for use with a 32.768 k crystal, enabling use of Timer/Counter0 as a Real Time Clock (RTC). Timer/Counter0 has its own prescale Timer/Counters 1 and 2 have individual prescaling selection from the same 10-bit prescaling timer. These Timer/Counte can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which trigge the counting.

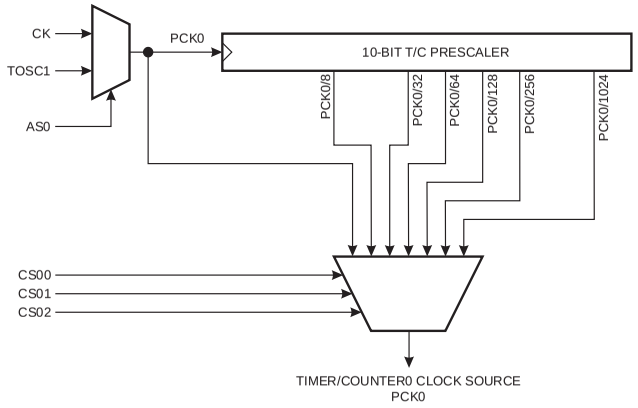
Timer/Counter Prescalers

Figure 28. Prescaler for Timer/Counter 1 and Timer/Counter2



For Timer/Counters 1 and 2, the four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK the CPU clock. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. F Timer/Counters 1 and 2, added selections as CK, external source and stop, can be selected as clock sources.

Figure 29. The Timer/Counter0 Prescaler



The clock source for Timer/Counter0 prescaler is named PCK0. PCK0 is by default connected to the main system clo
CK. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. By setting tl
AS0 bit in ASSR, Timer/Counter 0 prescaler is asynchronously clocked from the TOSC1 pin. This enables use
Timer/Counter0 as a Real Time Clock (RTC). A crystal can be connected between the TOSC1 and TOSC2 pins to serve
an independent clock source for Timer/Counter0. This oscillator is optimized for use with a 32.768 kHz crystal.

8-bit Timer/Counters T/C0 and T/C2

Figure 30 shows the block diagram for Timer/Counter0.

Figure 30. Timer/Counter0 Block Diagram

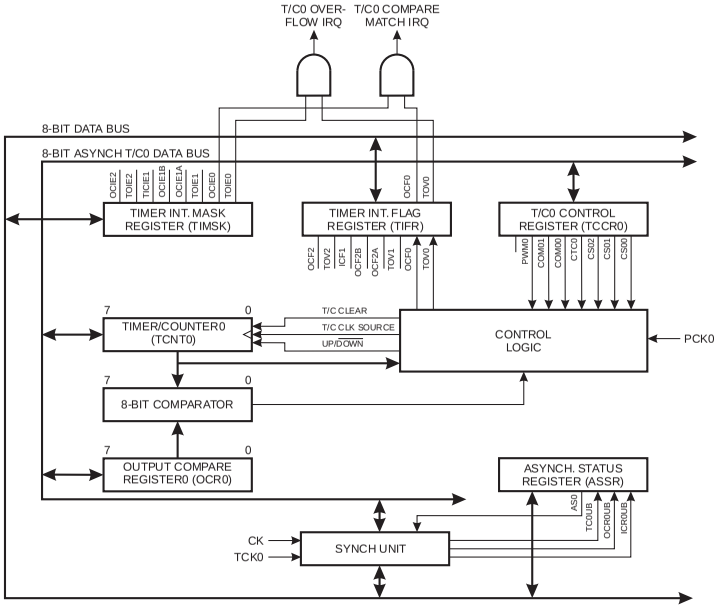
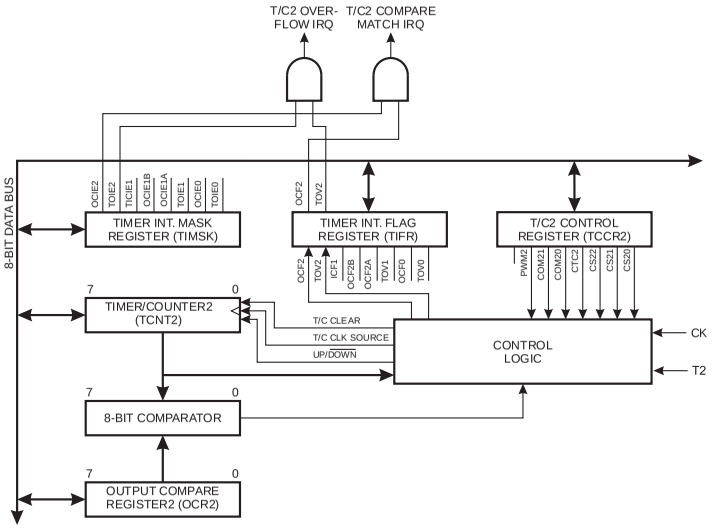


Figure 31. Timer/Counter2 Block Diagram



Note: Figure 31 shows the block diagram for Timer/Counter2.

The 8-bit Timer/Counter0 can select clock source from PCK0 or prescaled PCK0. The 8-bit Timer/Counter2 can sele
clock source from CK, prescaled CK, or an external pin. Both Timer/Counters can be stopped as described in the specific
tion for the Timer/Counter Control Registers - TCCR0 and TCCR2.

The different status flags (overflow, compare match and capture event) are found in the Timer/Counter Interrupt Flag Re
ister - TIFR. Control signals are found in the Timer/Counter Control Registers - TCCR0 and TCCR2. The interru
enable/disable settings are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter2 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU.
assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least o
internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counters feature a high resolution and a high accuracy usage with the lower prescaling opportunities. Si
ilarly, the high prescaling opportunities make these units useful for lower speed functions or exact timing functions w
infrequent actions.

Both Timer/Counters support two Output Compare functions using the Output Compare Registers - OCR0 and OCR2
the data source to be compared to the Timer/Counter contents. The Output Compare functions include optional clearing
the counter on compare match, and action on the Output Compare Pins - PB4(OC0/PWM0) and PB7(OC2/PWM2) -
compare match.

Timer/Counter0 and 2 can also be used as 8-bit Pulse Width Modulators. In this mode the Timer/Counter and the outp
compare register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 40 for a detailed descripti
on this function.

ITALIAN TECHNOLOGY grifo® abaco bus

Timer/Counter0 Control Register - TCCR0

Bit	7	6	5	4	3	2	1	0	
33 ($53)	-	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 Control Register - TCCR2

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	TCCR2
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - Res: Reserved Bit**

This bit is a reserved bit in the ATmega603/103 and always reads as zero.

- **Bit 6 - PWM0 / PWM2: Pulse Width Modulator Enable**

When set (one) this bit enables PWM mode for Timer/Counter0 or Timer/Counter2. This mode is described on page 40.

- **Bits 5,4 - COM01, COM00 / COM21, COM20: Compare Output Mode, bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter Any output pin actions affect pins PB4(OC0/PWM0) or PB7(OC2/PWM2). Since this is an alternative function to an I/ port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown Table 11.

Table 11. Compare Mode Select

COMn1	COMn0	Description
0	0	Timer/Counter disconnected from output pin OCn/PWMn
0	1	Toggle the OCn/PWMn output line.
1	0	Clear the OCn/PWMn output line (to zero).
1	1	Set the OCn/PWMn output line (to one).

Note: n = 0 or 2
In PWM mode, these bits have a different function. Refer to Table 14 for a detailed description.

- **Bit 3 - CTC0 / CTC2: Clear Timer/Counter on Compare match**

When the CTC0 or CTC2 control bit is set (one), the Timer/Counter is reset to $00 in the CPU clock cycle after a compa match. If the control bit is cleared, the Timer continues counting and is unaffected by a compare match. Since the compa match is detected in the CPU clock cycle following the match, this function will behave differently when a prescaling high than 1 is used for the timer. When a prescaling of 1 is used, and the compare register is set to C, the timer will count as fo lows if CTC0/2 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 1, ...

In PWM mode, this bit has no effect.

- **Bits 2,1,0 - CS02, CS01, CS00 / CS22, CS21, CS20: Clock Select bits 2,1 and 0**

The Clock Select2 bits 2,1 and 0 define the prescaling source of the Timer/Counter.

Table 12. Timer/Counter0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Timer/Counter0 is stopped.
0	0	1	PCK0
0	1	0	PCK0/8
0	1	1	PCK0/32
1	0	0	PCK0/64
1	0	1	PCK0/128
1	1	0	PCK0/256
1	1	1	PCK0/1024

Table 13. Timer/Counter2 Prescale Select

CS22	CS21	CS20	Description
0	0	0	Timer/Counter2 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin PD7(T2), falling edge
1	1	1	External Pin PD7(T2), rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the C CPU clock. If the external pin modes are used for Timer/Counter2, transitions on PD7/(T2) will clock the counter even if t pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
$32 ($42)	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 - TCNT2

Bit	7	6	5	4	3	2	1	0	
$24 ($44)	MSB							LSB	TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

These 8-bit registers contains the value of the Timer/Counters.

Both Timer/Counters are realized as up or up/down (in PWM mode) counters with read and write access. If t Timer/Counter is written to and a clock source is selected, it continues counting in the timer clock cycle after it is preset w the written value.

Timer/Counter0 Output Compare Register - OCR0

Bit	7	6	5	4	3	2	1	0	
$31 ($51)	MSB							LSB	OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 Output Compare Register - OCR2

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	MSB							LSB	OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The output compare registers are 8-bit read/write registers.

The Timer/Counter Output Compare Registers contain the data to be continuously compared with the Timer/Counte Actions on compare matches are specified in TCCR0 and TCCR2. A compare match does only occur if the Timer/Count counts to the OCR value. A software write that sets the Timer/Counter and Output Compare Register to the same val does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Timer/Counter 0 and 2 in PWM mode

When the PWM mode is selected, the Timer/Counter and the Output Compare Register - OCR0 or OCR2 form an 8-b free-running, glitch-free and phase correct PWM with outputs on the PB4(OC0/PWM0) or PB7(OC2/PWM2) pin. T Timer/Counter acts as an up/down counter, counting up from $00 to $FF, where it turns and counts down again to ze before the cycle is repeated. When the counter value matches the contents of the Output Compare register, t PB4(OC0/PWM0) or PB7(OC2/PWM2) pin is set or cleared according to the settings of the COM01/COM00 COM21/COM20 bits in the Timer/Counter Control Registers TCCR0 and TCCR2. Refer to Table 14 for details.

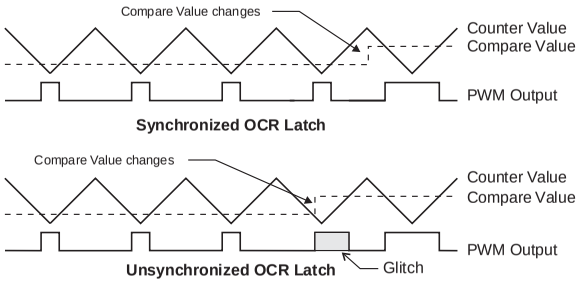
Table 14. Compare Mode Select in PWM Mode

COMn1	COMn0	Effect on Compare/PWM Pin
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note: n = 0 or 2

Note that in PWM mode, the Output Compare register is transferred to a temporary location when written. The value latched when the Timer/Counter reaches $FF. This prevents the occurrence of odd-length PWM pulses (glitches) in t event of an unsynchronized OCR0 or OCR2 write. See Figure 32 for an example.

Figure 32. Effects on Unsynchronized OCR Latching



During the time between the write and the latch operation, a read from OCR0 or OCR2 will read the contents of the temp rary location. This means that the most recently written value always will read out of OCR0/2

When the OCR register (not the temporary register) is updated to $00 or $FF, the PWM output changes to low or hi immediately according to the settings of COM21/COM20 or COM11/COM10. This is shown in Table 15:

Table 15. PWM Outputs OCRn = $00 or $FF

COMn1	COMn0	OCRn	Output PWMn
1	0	$00	L
1	0	$FF	H
1	1	$00	H
1	1	$FF	L

Note: n = 0 or 2

In PWM mode, the Timer Overflow Flag, TOV0 or TOV2, is set when the counter advances from $00. Timer Overflo Interrupt0 and 2 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV0 or TOV2 is set provid that Timer Overflow Interrupt and global interrupts are enabled. This does also apply to the Timer Output Compare fla and interrupts.

The frequency of the PWM will be Timer Clock Frequency divided by 510.

Asynchronous Status Register - ASSR

Bit	7	6	5	4	3	2	1	0	
$30 ($50)	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7..4 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always reads as zero.

- **Bit 3 - AS0: Asynchronous Timer/Counter0**

When set(one) Timer/Counter0 is clocked from the TOSC1 pin. When cleared (zero) Timer/Counter0 is clocked from t internal system clock, CK. When the value of this bit is changed the contents of TCNT0 might get corrupted.

- **Bit 2 - TCN0UB: Timer/Counter0 Update Busy**

When Timer/Counter0 operates asynchronously and TCNT0 is written, this bit becomes set (one). When TCNT0 has be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th TCNT0 is ready to be updated with a new value.

- **Bit 1 - OCR0UB: Output Compare Register0 Update Busy**

When Timer/Counter0 operates asynchronously and OCR0 is written, this bit becomes set (one). When OCR0 has be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th OCR0 is ready to be updated with a new value.

- **Bit 0 - TCR0UB: Timer/Counter Control Register0 Update Busy**

When Timer/Counter0 operates asynchronously and TCCR0 is written, this bit becomes set (one). When TCCR0 has be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th TCCR0 is ready to be updated with a new value.

If a write is performed to any of the three Timer/Counter0 registers while its update busy flag is set (one), the updated val might get corrupted and cause an unintentional interrupt to occur.

When reading TCNT0, OCR0 and TCCR0, there is a difference in result. When reading TCNT0, the actual timer value read. When reading OCR0 or TCCR0, the value in the temporary storage register is read.

Asynchronous Operation of Timer/Counter0

When Timer/Counter0 operates synchronously, all operations and timing are identical to Timer/Counter2. During asynchr nous operation, however, some considerations must be taken.

- WARNING: When switching between asynchronous and synchronous clocking of Timer/Counter0, the timer registers, TCNT0, OCR0 and TCCR0 might get corrupted. Safe procedure for switching clock source:
  1. Disable the timer 0 interrupts OCIE0 and TOIE0.
  2. Select clock source by setting ASO as appropriate.
  3. Write new values to TCNT0, OCR0 and TCCR0.
  4. If switching to asynchronous operation: Wait for TCNT0UB, OCR0UB and TCR0UB to be cleared.
  5. Enable interrupts if needed.
- The oscillator is optimized for use with a 32,768Hz watch crystal. An external clock signal applied to this pin goes through the same amplifier having a bandwidth of 256kHz. The external clock signal should therefore be in the interval 0Hz - 256kHz. The frequency of the clock signal applied to the TOSC1 pin must be lower than one fourth of the CPU main clock frequency. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled.
- When writing to one of the registers TCNT0, OCR0, or TCCR0, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporar register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g. writing to TCNT0 does not disturb an OCR0 write in progress. To detect that a transfer t the destination register has taken place, a Asynchronous Status Register - ASSR has been implemented.
- When entering Power Save mode after having written to TCNT0, OCR0 or TCCR0, the user must wait until the written register has been updated if Timer/Counter0 is used to wake up the device. Otherwise, the MCU will go to sleep befor the changes have had any effect. This is extremely important if the output compare0 interrupt is used to wake up the device; Output compare is disabled during write to OCR0 or TCNT0. If the write cycle is not finished (i.e. the user goes t sleep before the OCR0UB bit returns to zero), the device will never get a compare match and the MCU will not wake u
- If Timer/Counter0 is used to wake up the device from Power Save mode, precautions must be taken if the user wants t re-enter Power Save mode; The interrupt logic needs one TOSC1 cycle to get reset. If the time between wake up and r entering Power Save mode is less than one TOSC1 cycle, the interrupt will not occur and the device will fail to wake up. the user is in doubt whether the time before re-entering Power Save is sufficient, the following algorithm can be used t ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR0, TCNT0 or OCR0
  2. Wait until the corresponding Update Busy flag in ASSR returns to zero.
  3. Enter Power Save mode
- When asynchronous operation is selected, the 32 kHz oscillator for Timer/Counter0 is always running, except in powe down mode. After a power up reset or wake-up from power down, the user should be aware of the fact that this oscillat might take as long as one second to stabilize. Therefore, the content of all Timer/Counter0 registers must be considere

lost after a wake-up from power down, due to the unstable clock signal. The user is advised to wait for at least one second before using Timer/Counter0 after power-up or wake-up from power down.

- Description of wake up from power save mode when the timer is clocked asynchronously: When the interrupt condition met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by a least one before the processor can read the counter value. To execute the corresponding Timer/Counter0 interrupt routine, the global interrupt bit in SREG must have been set. Otherwise, the part will still wake up from power down, bu continues to execute the sleep command. The interrupt flags are updated 3 processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and th interrupt routine has not started yet.
- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes 3 processc cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the interrupt flag. The output compare pin is changed on the timer clock, and is not synchronized to the processor clock.

16-bit Timer/Counter1

Figure 33 shows the block diagram for Timer/Counter1.

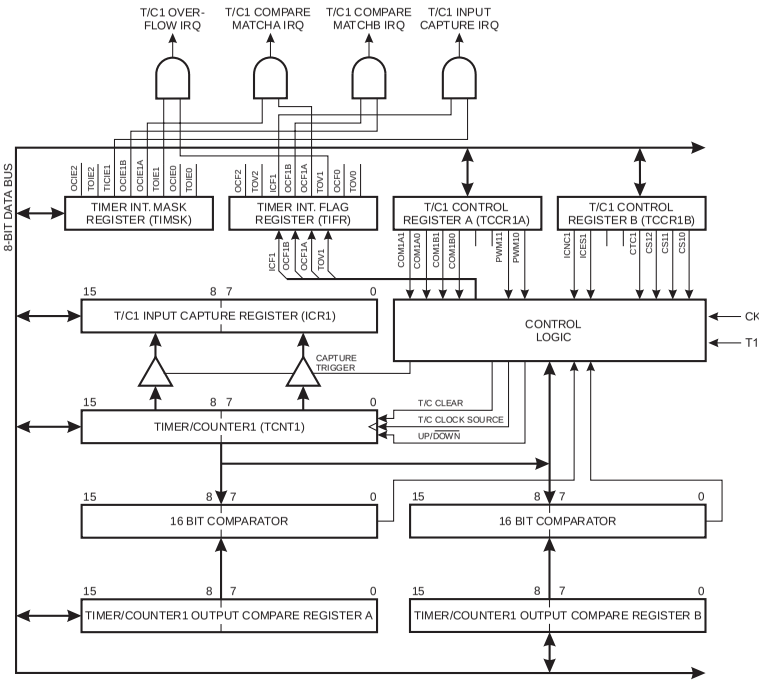
The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopp as described in the specification for the Timer/Counter1 Control Register - TCCR1B. The different status flags (overflo compare match and capture event) are found in the Timer/Counter Interrupt Flag Register - TIFR. Control signals are fou in the Timer/Counter1 Control Registers - TCCR1A and TCCR1B. The interrupt enable/disable settings for Timer/Counte are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least o internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportu ties. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timi functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B - OCR1A a OCR1B as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions inclu optional clearing of the counter on compareA match, and actions on the Output Compare pins on both compare matches

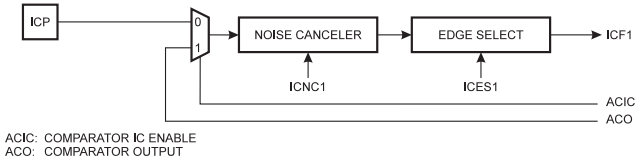
Figure 33. Timer/Counter1 Block Diagram



Timer/Counter1 can also be used as a 8, 9 or 10-bit Pulse With Modulator. In this mode the counter and th OCR1A/OCR1B registers serve as a dual glitch-free stand-alone PWM with centered pulses. Refer to page 49 for detailed description on this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Captu Register - ICR1, triggered by an external event on the Input Capture Pin - PD4/(IC1). The actual capture event settin are defined by the Timer/Counter1 Control Register - TCCR1B. In addition, the Analog Comparator can be set to trigg the Input Capture. Refer to the paragraph, “The Analog Comparator”, for details on this. The ICP pin logic is shov in Figure 34.

Figure 34. ICP Pin Schematic Diagram



If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples, a all 4 must be equal to activate the capture flag.

Timer/Counter1 Control Register A - TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..6 - COM1A1, COM1A0: Compare Output Mode1A, bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a compare match in Timer/Counter Any output pin actions affect pin OC1A - Output CompareA pin 1. This is an alternative function to an I/O port, and the co responding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 16

- **Bits 5..4 - COM1B1, COM1B0: Compare Output Mode1B, bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter Any output pin actions affect pin OC1B - Output CompareB. Since this is an alternative function to an I/O port, the corr sponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

Table 16. Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Note: X = A or B.
In PWM mode, these bits have a different function. Refer to Table 17 for a detailed description.

- **Bits 3..2 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and always read zero.

- **Bits 1..0 - PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 19. This mode is described on page 49.

Table 17. PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled
0	1	Timer/Counter1 is an 8-bit PWM
1	0	Timer/Counter1 is a 9-bit PWM
1	1	Timer/Counter1 is a 10-bit PWM

ITALIAN TECHNOLOGY grifo® abaco bus

Timer/Counter1 Control Register B - TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is tri gered at the first rising/falling edge sampled on the input capture pin PD4(IC1) as specified. When the ICNC1 bit is s (one), four successive samples are measures on PD4(IC1), and all samples must be high/low according to the input ca ture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 - ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - the falling edge of the input capture pin - PD4(IC1). While the ICES1 bit is set (one), the Timer/Counter1 contents are tran ferred to the Input Capture Register - ICR1 - on the rising edge of the input capture pin - PD4(IC1).

- **Bits 5, 4 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and always read zero.

- **Bit 3 - CTC1: Clear Timer/Counter1 on Compare Match**

When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000 in the clock cycle after a compareA match. the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match. Since the co pare match is detected in the CPU clock cycle following the match, this function will behave differently when a prescali higher than 1 is used for the timer. When a prescaling of 1 is used, and the compareA register is set to C, the timer v count as follows i CTC1 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...

In PWM mode, this bit has no effect.

- **Bits 2,1,0 - CS12, CS11, CS10: Clock Select1, bit 2,1 and 0**

The lock Select1 bits 2,1 and 0 define the prescaling source of Timer/Counter1.

Table 18. Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T1, falling edge
1	1	1	External Pin T1, rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the C CPU clock. If the external pin modes are used for Timer/Counter1, transitions on PD6/(T1) will clock the counter even if t pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter1 - TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	MSB								TCNT1H
$2C ($4C)								LSB	TCNT1L
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low byt are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit te porary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the ma program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during acce from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- **TCNT1 Timer/Counter1 Write:**

  When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CP writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits a written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be access first for a full 16-bit register write operation. When using Timer/Counter1 as an 8-bit timer, it is sufficient to write the l byte only.

- **TCNT1 Timer/Counter1 Read:**

  When the CPU reads the low byte TCNT1L, the data of TCNT1L is sent to the CPU and the data of the high by TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives t data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register re operation. When using Timer/counter1 as an 8-bit timer, it is sufficient to read the low byte only.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counte is written to and a clock source is selected, the Timer/Counter1 continues counting in the clock cycle after it is preset w the written value.

Timer/Counter1 Output Compare Register - OCR1AH and OCR1AL

Bit	15	14	13	12	11	10	9	8	
$2B	MSB								OCR1AH
$2A								LSB	OCR1AL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Timer/Counter1 Output Compare Register - OCR1BH and OCR1BL

Bit	15	14	13	12	11	10	9	8	
$29	MSB								OCR1BH
$28								LSB	OCR1BL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The output compare registers are 16-bit read/write registers.

The Timer/Counter1 Output Compare Registers contain the data to be continuously compared with Timer/Counter Actions on compare matches are specified in the Timer/Counter1 Control and Status register. A compare match does or occur if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the sar value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers - OCR1A and OCR1B - are 16-bit registers, a temporary register TEMP is used wh OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1A or OCR1BH, the data is temporarily stored in the TEMP register. When the CPU writes the low byte, OCR1AL or OCR1E the TEMP register is simultaneously written to OCR1AH or OCR1BH. Consequently, the high byte OCR1AH or OCR1E must be written first for a full 16-bit register write operation.

The TEMP register is also used when accessing TCNT1, and ICR1. If the main program and also interrupt routines perfor access to registers using TEMP, interrupts must be disabled during access from the main program.

Timer/Counter1 Input Capture Register - ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($37)	MSB								ICR1H
$26 ($36)								LSB	ICR1L
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting - ICES1) of the signal at the input capture pi PD4(IC1) - is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At t same time, the input capture flag - ICF1 - is set (one).

Since the Input Capture Register - ICR1 - is a 16-bit register, a temporary register TEMP is used when ICR1 is read ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU a the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, t CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit re ister read operation.

The TEMP register is also used when accessing TCNT1, OCR1A and OCR1B. If the main program and also interru routines perform access to registers using TEMP, interrupts must be disabled during access from the main program.

Timer/Counter1 in PWM mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A - OCR1A and the Output Compa Register1B - OCR1B, form a dual 8, 9 or 10-bit, free-running, glitch-free and phase correct PWM with outputs on t PB5(OC1A) and PB6(OC1B) pins. Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Tab 17), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the co tents of the 10 least significant bits of OCR1A or OCR1B, the PB5(OC1A)/PB6(OC1B) pins are set or cleared according the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer Table 20 for details.

Table 19. Timer TOP Values and PWM Frequency

PWM Resolution	Timer TOP value	Frequency
8-bit	$00FF (255)	$f_{TCK1}/510$
9-bit	$01FF (511)	$f_{TCK1}/1022$
10-bit	$03FF(1023)	$f_{TCK1}/2046$

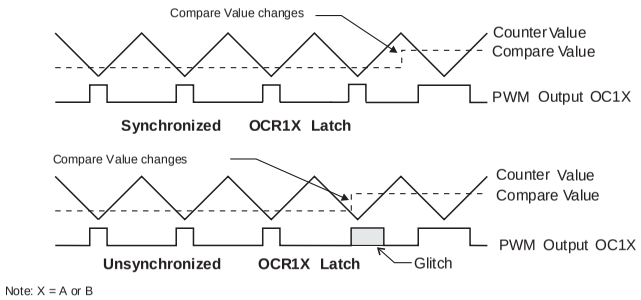
Table 20. Compare1 Mode Select in PWM Mode

COM1X1	COM1X0	Effect on OCX1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note: X = A or B

Note that in the PWM mode, the 10 least significant OCR1A/OCR1B bits, when written, are transferred to a temporary loc tion. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PW pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 35 for an example.

Figure 35. Effects on Unsynchronized OCR1 Latching



Note: X = A or B

During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the ter porary location. This means that the most recently written value always will read out of OCR1A/B

When the OCR1A/OCR1B contains $0000 or TOP, the output OC1A/OC1B is updated to low or high on the next compa match according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 21.

Table 21. PWM Outputs OCR1X = $0000 Or TOP

COM1X1	COM1X0	OCR1X	Output OC1X
1	0	$0000	L
1	0	TOP	H
1	1	$0000	H
1	1	TOP	L

Note: X = A or B

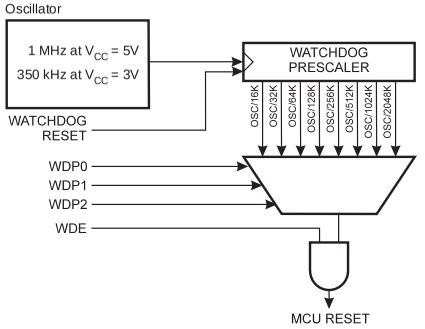
In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter advances from $0000. Timer Overflow Interrup operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overfl Interrupt1 and global interrupts are enabled. This does also apply to the Timer Output Compare1 flags and interrupts.

Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator. By controlling the Watchdog Timer prescaler, tl Watchdog reset interval can be adjusted as shown in Table 22. See characterization data for typical values at other V levels. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. From the Watchdog is reset, eight differe clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchd reset, the ATmega603/103 resets and executes from the reset vector. For timing details on the Watchdog reset, refer page 28.

To prevent unintentional disabling of the watchdog, a special turn-off procedure must be followed when the watchdog disabled. Refer to the description of the Watchdog Timer Control Register for details.

Figure 36. Watchdog Timer



Watchdog Timer Control Register - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..5 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always read as zero.

- **Bit 4 - WDTOE: Watch Dog Turn Off Enable**

This bit must be set (one) when the WDE bit is cleared, Otherwise, the watchdog will not be disabled. Once set, hardwa will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure.

- **Bit 3 - WDE: Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer functi is disabled. WDE can only be cleared if the WDTOE bit is set (one). To disable an enabled watchdog timer, the followi procedure must be followed:

1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

- **Bits 2..0 - WDP2, WDP1, WDP0: Watch Dog Timer Prescaler 2, 1 and 0**

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. Tl different prescaling values and their corresponding Time-out Periods are shown in Table 22.

Table 22. Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator cycles	Typical time-out at Vcc = 3.0V	Typical time-out at Vcc = 5.0V
0	0	0	16K cycles	47 ms	15 ms
0	0	1	32K cycles	94 ms	30 ms
0	1	0	64K cycles	0.19 s	60 ms
0	1	1	128K cycles	0.38 s	0.12 s
1	0	0	256K cycles	0.75 s	0,24 s
1	0	1	512K cycles	1.5 s	0.49 s
1	1	0	1,024K cycles	3.0 s	0.97 s
1	1	1	2,048K cycles	6.0 s	1.9 s

Note: The frequency of the watchdog oscillator is voltage dependent as shown in the Electrical Characteristics section. The WDR - Watchdog Reset - instruction should always be executed before the Watchdog Timer is enabled. This ensures tha the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the watchdog timer may not start counting from zero.

EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space.

The write access time is in the range of 2.5 - 4ms, depending on the V_{CC} voltages. A self-timing function lets the user so ware detect when the next byte can be written. A special EEPROM Ready interrupt can be set to trigger when th EEPROM is ready to accept new data.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description the EEPROM control register for details on this.

When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed. When it read, the CPU is halted for 4 clock cycles.

EEPROM Address Register - EEARH, EEARL

Bit	15	14	13	12	11	10	9	8	
$1F ($3F)	-	-	-	-	EEAR11	EEAR10	EEAR9	EEAR8	EEARH
$1E ($3E)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The EEPROM Address Registers - EEARH and EEARL specify the EEPROM address in the 2K/4K-byte EEPROM spac The EEPROM data bytes are addressed linearly between 0 and 2047/4095. The ATmega603 has an EEPROM addre space of 2K, and the EEAR11 I/O bit is read-only with initial value of 0.

EEPROM Data Register - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - EEDR7..0: EEPROM Data:**

For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address giv by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at t address given by EEAR.

EEPROM Control Register - EECR

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always be read as zero.

- **Bit 3 - EERIE: EEPROM Ready Interrupt Enable**

When the I bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the inte rupt is disabled. The EEPROM Ready interrupt constantly generates an interrupt request when EEWE is cleared (zero).

- **Bit 2 - EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set(or setting EEWE will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effe When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the descripti of the EEWE bit for a EEPROM write procedure.

- **Bit 1 - EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set u the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical one is writt to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing th EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEAR (optional)
3. Write new EEPROM data to EEDR (optional)
4. Write a logical one to the EEMWE bit in EECR
5. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

Caution: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable w time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR and EEDR re ister will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt fl cleared during the 4 last steps to avoid these problems.

When the write access time (typically 2.5 ms at V_{CC} = 5V or 4 ms at V_{CC} = 2.7V) has elapsed, the EEWE bit is clear (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE h been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 - EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in th EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in t EEDR register. The EEPROM read access takes one instruction there is no need to poll the EERE bit. When EERE h been set, the CPU is halted for four cycles before the next instruction is executed.

GPC® AM4 Rel. 3.00
Pagina B-23
ITALIAN TECHNOLOGY grifo®

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress when new data address is written to the EEPROM I/O registers, the write operation will be interrupted, and the result is undefined.

Prevent EEPROM Corruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and t EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM, and the sar design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequen to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions inc rectly, if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

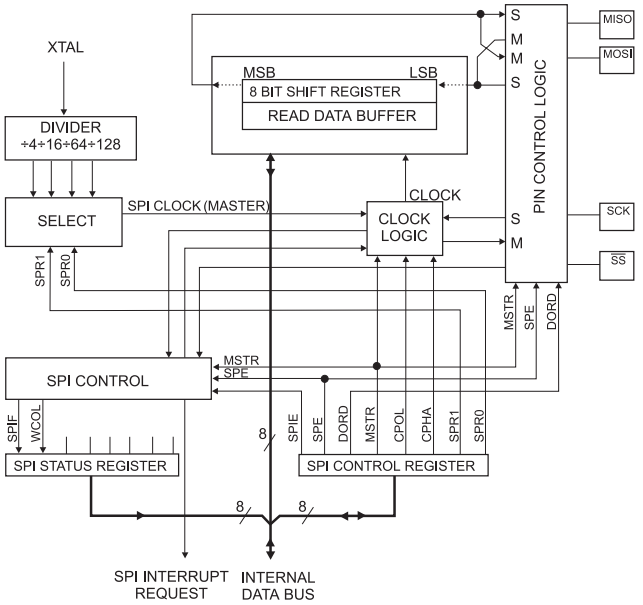
1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low V_{CC} Reset Protection circuit, often referred to as a Brown-Out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low voltage detection.
2. Keep the AVR core in Power Down Sleep Mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory can not be updated by the CPU, and will not be subject to corruption.

Serial Peripheral Interface - SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega603/103 a peripheral devices or between several AVR devices. The ATmega603/103 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode (Slave Mode Only)

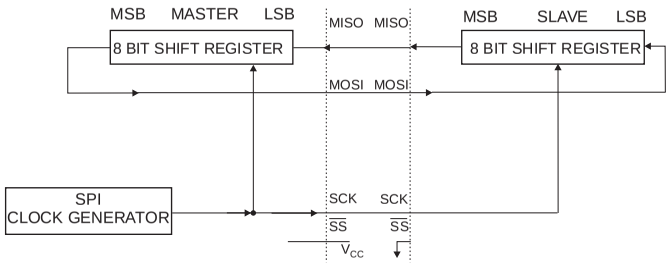
Figure 37. SPI Block Diagram



The interconnection between master and slave CPUs with SPI is shown in Figure 38. The PB1(SCK) pin is the clock outp in the master mode and is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts t SPI clock generator, and the data written shifts out of the PB2(MOSI) pin and into the PB2 (MOSI) pin of the slave CP After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If the SPI interrupt enat bit (SPIE) in the SPCR register is set, an interrupt is requested. The Slave Select input, PB0($\overline{SS}$), is set low to select individual slave SPI device. The two shift registers in the Master and the Slave can be considered as one distributed 16- circular shift register. This is shown in Figure 38. When data is shifted from the master to the slave, data is also shifted the opposite direction, simultaneously. This means that during one shift cycle, data in the master and the slave a interchanged.

GPC® AM4 Rel. 3.00
abaco bus
grifo®
ITALIAN TECHNOLOGY

Figure 38. SPI Master-Slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that chara ters to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receivi data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifte in. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and $\overline{SS}$ pins is overridden according to the followi table:

Table 23. SPI Pin Overrides

PIN	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

Note: See “Alternate Functions of Port B” on page 78 for a detailed description and how to define the direction of the user defined SPI pins.

$\overline{SS}$ Pin Functionality

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ configured as an output, the pin is a general output pin which does not affect the SPI system. If $\overline{SS}$ is configured as input, it must be hold high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI configured as master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting t SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a slave. As a result of the SPI becoming a slave, the MOSI and SCK pins become inputs.
2. The SPIF flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

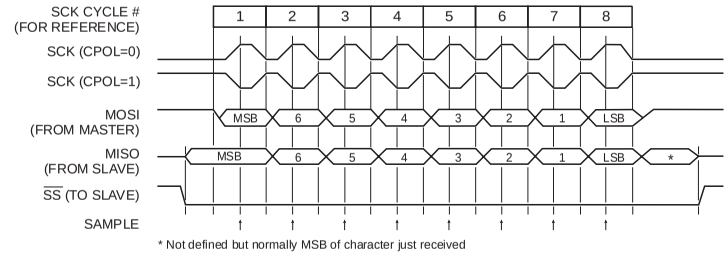
Thus, when interrupt-driven SPI transmittal is used in master mode, and there exists a possibility that $\overline{SS}$ is driven low, t interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it mu be set by the user to re-enable SPI master mode.

When the SPI is configured as a slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MIS becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs, a the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ p is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately a both data received and data sent must be considered as lost.

Data Modes

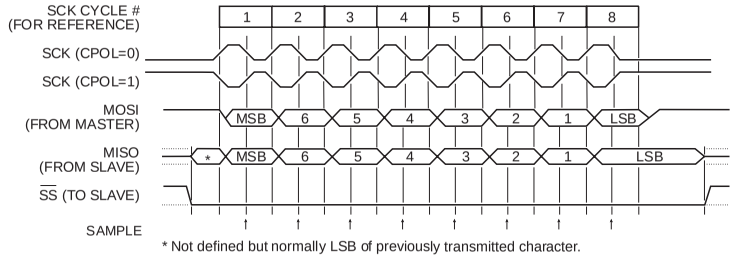
There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control b CPHA and CPOL. The SPI data transfer formats are shown in Figure 39 and Figure 40.

Figure 39. SPI Transfer Format with CPHA = 0 and DORD = 0



* Not defined but normally MSB of character just received

Figure 40. SPI Transfer Format with CPHA = 1and DORD = 0



* Not defined but normally LSB of previously transmitted character.

SPI Control Register - SPCR

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR register is set and the global interrupts are enable

- **Bit 6 - SPE: SPI Enable**

When the SPE bit is set (one), the SPI is enabled and $\overline{SS}$, MOSI, MISO and SCK are connected to pins PB0, PB1, PB2 a PB3.

- **Bit 5 - DORD: Data Order**

When the DORD bit is set (one), the LSB of the data word is transmitted first.

When the DORD bit is cleared (zero), the MSB of the data word is transmitted first.

- **Bit 4 - MSTR: Master/Slave Select**

This bit selects Master SPI mode when set (one), and Slave SPI mode when cleared (zero). If $\overline{SS}$ is configured as an inp and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have set MSTR to re-enable SPI master mode.

- **Bit 3 - CPOL: Clock Polarity**

When this bit is set (one), SCK is high when idle. When CPOL is cleared (zero), SCK is low when idle. Refer to Figure and Figure 40 for additional information.

- **Bit 2 - CPHA: Clock Phase**

Refer to Figure 39 or Figure 40 for the functionality of this bit.

- **Bits 1,0 - SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave. Th relationship between SCK and the CPU Clock frequency f_{cl} is shown in the following table:

Table 24. Relationship Between SCK and the Oscillator Frequency

SPR1	SPR0	SCK Frequency
0	0	f_{cl} / 4
0	1	f_{cl} / 16
1	0	f_{cl} / 64
1	1	f_{cl} / 128

Note: Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled.

SPI Status Register - SPSR

Bit	7	6	5	4	3	2	1	0	
$0E	SPIF	WCOL	-	-	-	-	-	-	SPSR
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) a global interrupts are enabled. SPIF is cleared by hardware when executing the corresponding interrupt handling vect Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set (one), then accessing the S Data Register (SPDR).

- **Bit 6 - WCOL: Write Collision flag**

The WCOL bit is set if the SPI data register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) a cleared (zero) by first reading the SPI Status Register with WCOL set (one), and then accessing the SPI Data Register.

- **Bit 5..0 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always read as zero.

SPI Data Register - SPDR

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	X	X	X	X	X	X	X	X	Undefined

The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift regist Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be rea

UART

The ATmega603/103 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver a Transmitter (UART). The main features are:

- Baud rate generator that can generate a large number of baud rates (bps)
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data Transmission

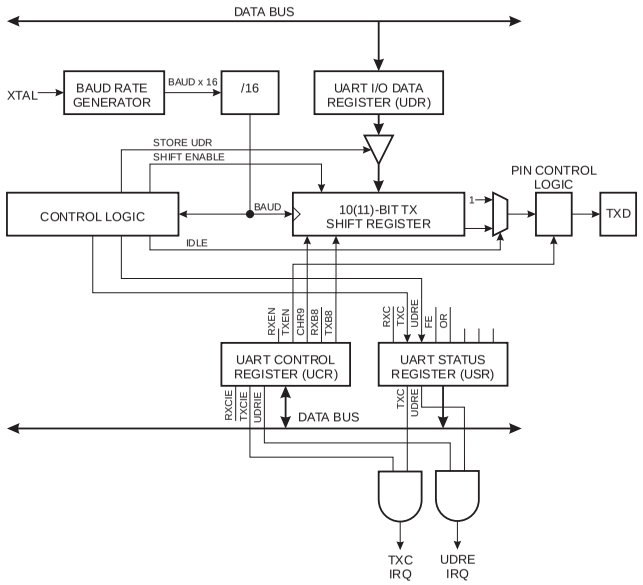
A block schematic of the UART transmitter is shown in Figure 41.

Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferr from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The sh register is loaded when the stop bit of the character currently being transmitted has been shifted out.

If the 10(11)-bit Transmitter shift register is empty, data is transferred from UDR to the shift register. At this time the UDF (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready receive the next character. Writing to UDR clears UDRE. At the same time as the data is transferred from UDR to t 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9 bit data word selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Trar mit shift register.

Figure 41. UART Transmitter

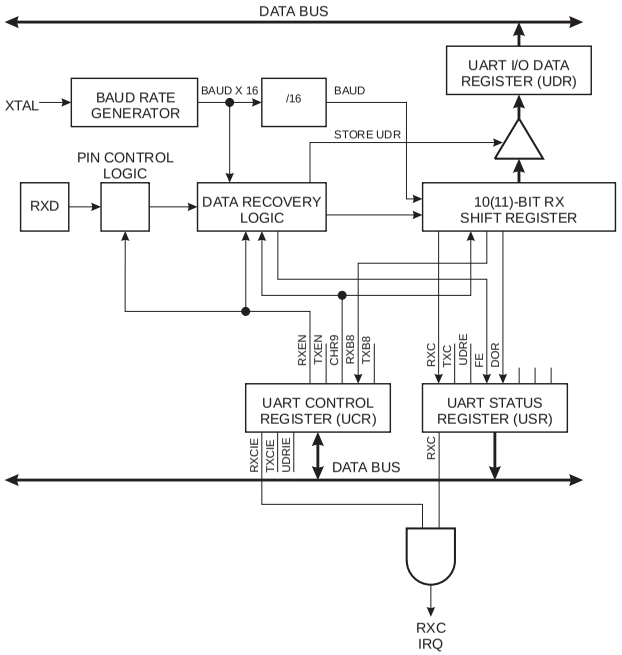


On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin, f
lowed by the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has be
written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to se
when the stop bit is shifted out, the UDRE flag will remain set. In this case, after the stop bit has been present on TXD f
one bit length, the TX Complete Flag, TXC, in USR is set.

The TXEN bit in UCR enables the UART transmitter when set (one). When this bit is cleared (zero), the PE1 pin can
used for general I/O. When TXEN is set, the UART Transmitter will be connected to PE1, which is forced to be an outp
pin regardless of the setting of the DDE1 bit in DDRE.

Data Reception

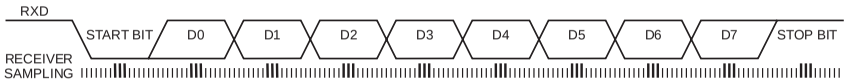
Figure 42. UART Receiver



The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is id
one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence
initiated. Let sample 1 denote the first zero-sample. Following the 1 to 0-transition, the receiver samples the RXD pin
sample 8, 9, and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noi
spike and the receiver starts looking for the next 1 to 0-transition.

If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are al
sampled at samples 8, 9, and 10. The logical value found in at least two of the three samples is taken as the bit value. .
bits are shifted into the transmitter shift register as they are sampled. Sampling of an incoming character is shown in Figu
43.

Figure 43. Sampling Received Data



ITALIAN TECHNOLOGY grifo® abaco bus

When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or mo samples are logical zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set when the received byte transferred to UDR. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors. F is cleared when UDR is read.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and tl RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received dat When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register accessed. If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR loaded with bit 9 in the Transmit shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been accessed since the last receive, the OverRun (OR) fl in USR is set. This means that the new data transferred to the shift register could not be transferred to UDR and is lost. Tl OR bit is buffered, and is available when the valid data byte in UDR has been read. The user should always check the C after reading from the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PE0 pin can be us as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PE0, which is forced to be an input p regardless of the setting of the DDE0 bit in DDRE. When PE0 is forced to input by the UART, the PORTE0 bit can still l used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9-bit long plus start and stop bi The 9th data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a tran mission is initated by writing to the UDR register. The 9th

UART Control

UART I/O Data Register - UDR

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the registe the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

UART Status Register - USR

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC	TXC	UDRE	FE	OR	-	-	-	USR
Read/Write	R	R/W	R	R	R	R	R	R	
Initial value	0	0	1	0	0	0	0	0	

The USR register is a read-only register providing information on the UART Status.

- **Bit 7 - RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regar less of any detected framing errors. When the RXCIE bit in UCR is set, the UART Receive Complete interrupt will l executed when RXC is set(one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UAF Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once tl interrupt routine terminates.

- **Bit 6 - TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out a no new data has been written to the UDR. This flag is especially useful in half-duplex communications interfaces, where transmitting application must enter receive mode and free the communications bus immediately after completing tl transmission.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is clear (zero) by writing a logical one to the bit.

- **Bit 5 - UDRE: UART Data Register Empty**

This bit is set (one) when a character written to UDR is transferred to the Transmit shift register. Setting of this bit indicat that the transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routi must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the transmitter is ready.

- **Bit 4 - FE: Framing Error**

This bit is set if a Framing Error condition is detected, i.e. when the stop bit of an incoming character is zero.

The FE bit is cleared when the stop bit of received data is one.

- **Bit 3 - OR: OverRun**

This bit is set if an Overrun condition is detected, i.e. when a character already present in the UDR register is not re before the next character is transferred from the Receiver Shift register. The OR bit is buffered, which means that it is v be set once the valid data still in UDRE is read.

The OR bit is cleared (zero) when data is received and transferred to UDR.

- **Bits 2..0 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always read as zero.

UART Control Register - UCR

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	1	0	

- **Bit 7 - RXCIE: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete interrupt routine to be execut provided that global interrupts are enabled.

- **Bit 6 - TXCIE: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete interrupt routine to be execut provided that global interrupts are enabled.

- **Bit 5 - UDRIE: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty interrupt routine to executed provided that global interrupts are enabled.

- **Bit 4 - RXEN: Receiver Enable**

This bit enables the UART receiver when set (one). When the receiver is disabled, the TXC, OR and FE status flags cann become set. If these flags are set, turning off RXEN does not cause them to be cleared.

- **Bit 3 - TXEN: Transmitter Enable**

This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, t transmitter is not disabled before the character in the shift register plus any following character in UDR has been co pletely transmitted.

- **Bit 2 - CHR9: 9 Bit Characters**

When this bit is set (one) transmitted and received characters are 9 bit long plus start and stop bits. The 9th bit is read a written by using the RXB8 and TXB8 bits in UCR, respectively. The 9th data bit can be used as an extra stop bit or a par bit.

- **Bit 1 - RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the 9th data bit of the received character.

- **Bit 0 - TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the 9th data bit in the character to be transmitted.

Baud Rate Generator

The baud rate generator is a frequency divider which generates baud rates according to the following equation:

$$BAUD = \frac{f_{CK}}{16(UBRR + 1)}$$

- BAUD = Baud Rate
- f_{CK} = CPU Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0 - 255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings Table 25. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. UBF values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table. Howeve using baud rates that have more than 1% error is not recommended. High error ratings give less noise resistance.

Table 25. UBRR Settings at Various CPU Frequencies

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBRR=	**25**	**0.2**	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**63**	**0.0**
4800	UBRR=	**12**	**0.2**	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**31**	**0.0**
9600	UBRR=	6	7.5	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**15**	**0.0**
14400	UBRR=	3	7.8	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	**7**	**0.0**
28800	UBRR=	1	7.8	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	4	6.3
38400	UBRR=	1	22.9	UBRR=	**2**	**0.0**	UBRR=	2	7.8	UBRR=	**3**	**0.0**
57600	UBRR=	0	7.8	UBRR=	**1**	**0.0**	UBRR=	1	7.8	UBRR=	2	12.5
76800	UBRR=	0	22.9	UBRR=	1	33.3	UBRR=	1	22.9	UBRR=	**1**	**0.0**
115200	UBRR=	0	84.3	UBRR=	**0**	**0.0**	UBRR=	0	7.8	UBRR=	0	25.0

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBRR=	**84**	**0.4**	UBRR=	**95**	**0.0**	UBRR=	**103**	**0.2**	UBRR=	**119**	**0.0**
4800	UBRR=	**42**	**0.8**	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**59**	**0.0**
9600	UBRR=	**20**	**1.6**	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**29**	**0.0**
14400	UBRR=	**13**	**1.6**	UBRR=	**15**	**0.0**	UBRR=	16	2.1	UBRR=	**19**	**0.0**
19200	UBRR=	10	3.1	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**14**	**0.0**
28800	UBRR=	**6**	**1.6**	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	**9**	**0.0**
38400	UBRR=	4	6.3	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	7	6.7
57600	UBRR=	3	12.5	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	**4**	**0.0**
76800	UBRR=	2	12.5	UBRR=	**2**	**0.0**	UBRR=	2	7.8	UBRR=	3	6.7
115200	UBRR=	1	12.5	UBRR=	**1**	**0.0**	UBRR=	1	7.8	UBRR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBRR=	**191**	**0.0**	UBRR=	**207**	**0.2**	UBRR=	**239**	**0.0**	UBRR=	287	-
4800	UBRR=	**95**	**0.0**	UBRR=	**103**	**0.2**	UBRR=	**119**	**0.0**	UBRR=	**143**	**0.0**
9600	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**59**	**0.0**	UBRR=	**71**	**0.0**
14400	UBRR=	**31**	**0.0**	UBRR=	**34**	**0.8**	UBRR=	**39**	**0.0**	UBRR=	**47**	**0.0**
19200	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**29**	**0.0**	UBRR=	**35**	**0.0**
28800	UBRR=	**15**	**0.0**	UBRR=	16	2.1	UBRR=	**19**	**0.0**	UBRR=	**23**	**0.0**
38400	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**14**	**0.0**	UBRR=	**17**	**0.0**
57600	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	**9**	**0.0**	UBRR=	**11**	**0.0**
76800	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	7	6.7	UBRR=	**8**	**0.0**
115200	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	**4**	**0.0**	UBRR=	**5**	**0.0**

UART Baud Rate Register - UBRR

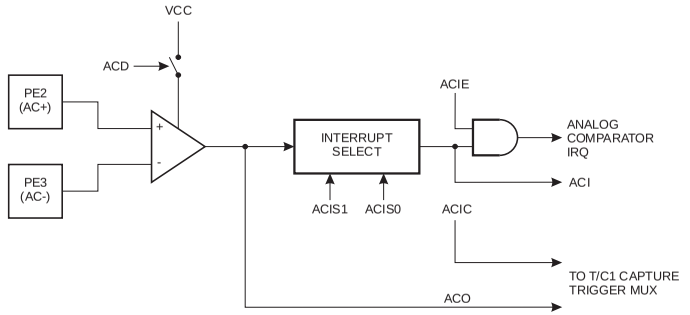
Bit	7	6	5	4	3	2	1	0	
$09 ($29)	**MSB**							**LSB**	**UBRR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UBRR is an 8-bit read/write register which specifies the UART Baud Rate according to the description on the previo page.

Analog Comparator

The analog comparator compares the input values on the positive input PE2 (AC+) and negative input PE3 (AC-). Wh the voltage on the positive input PE2 (AC+) is higher than the voltage on the negative input PE3 (AC-), the Analog Compa ator Output, ACO is set (one). The output of the comparator can be set to trigger the Timer/Counter1 Input Captu function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user c select Interrupt triggering on comparator output rise, fall, or toggle. A block diagram of the comparator and its surroundi logic is shown in Figure 44.

Figure 44. Analog Comparator Block Diagram



Analog Comparator Control and Status Register - ACSR

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	**ACD**	**-**	**ACO**	**ACI**	**ACIE**	**ACIC**	**ACIS1**	**ACIS0**	**ACSR**
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - ACD: Analog Comparator Disable**

When this bit is set(one), the power to the analog comparator is switched off. This bit can be set at any time to turn off tl analog comparator. This will reduce power consumption in active and idle mode. When changing the ACD bit, the Anal Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit changed.

- **Bit 6 - Res: Reserved bit**

This bit is a reserved bit in the ATmega603/103 and will always read as zero.

- **Bit 5 - ACO: Analog Comparator Output**

ACO is directly connected to the comparator output.

- **Bit 4 - ACI: Analog Comparator Interrupt Flag**

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Anal Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one the flag. Observe however, that if another bit in this register is modified using the SBI or CBI instruction, ACI will be clear if it has become set before the operation.

- **Bit 3 - ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activate When cleared (zero), the interrupt is disabled.

- **Bit 2 - ACIC: Analog Comparator Input Capture enable**

When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparat The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utili the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no cc nection between the analog comparator and the Input Capture function is given. To make the comparator trigger t Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

- **Bits 1,0 - ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings a shown in Table 26.

Table 26. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enat bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

Caution: Using the SBI or CBI instruction on other bits than ACI in this register, will write a one back into ACI if it is read set, thus clearing the flag.

Analog to Digital Converter

Feature list:

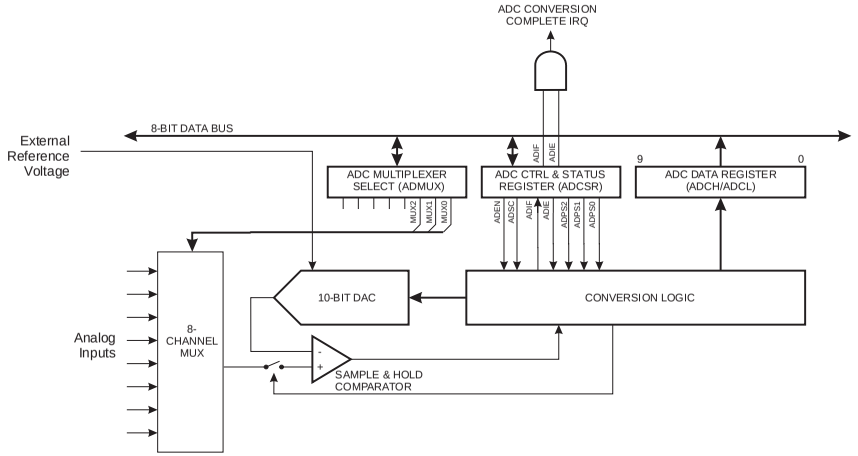
- 10-bit Resolution
- ±2 LSB absolute accuracy
- 0.5 LSB Integral Non-Linearity
- 70 - 280 µs conversion time
- Up to 14 kSPS
- 8 Multiplexed Input Channels
- Interrupt on ADC conversion complete.
- Sleep Mode Noise Canceler

The ATmega603/103 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog M tiplexer which allows each pin of Port F to be used as an input for the ADC. The ADC contains a Sample and Hold Amplif which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the AI is shown in Figure 45.

The ADC has two separate analog supply voltage pins, AV_{CC} and AGND. AGND must be connected to GND, and the vc age on AV_{CC} must not differ more than ± 0.3 V from V_{CC}. See the section “ADC Noise Canceling Techniques” on page on how to connect these pins.

An external reference voltage must be applied to the AREF pin. This voltage must be in the range AGND - AV_{CC}.

Figure 45. Analog to Digital Converter Block Schematic



Operation

The ADC operates in Single Conversion mode, and each conversion will have to be initiated by the user.

The ADC is enabled by writing a logical one to the ADC Enable bit, ADEN in ADCSR. The first conversion that is start after enabling the ADC, will be preceded by a dummy conversion to initialize the ADC. To the user, the only difference v be that this conversion takes 13 more ADC clock pulses than a normal conversion. (See Figure 48.)

A conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit will stay high as long as t conversion is in progress and be set to zero by hardware when the conversion is completed. If a different data channel selected while a conversion is in progress, the ADC will finish the current conversion before performing the chanr change.

As the ADC generates a 10-bit result, two data registers, ADCH and ADCL, must be read to get the result when the conve sion is complete. Special data protection logic is used to ensure that the contents of the data registers belong to the sar result when they are read. This mechanism works as follows:

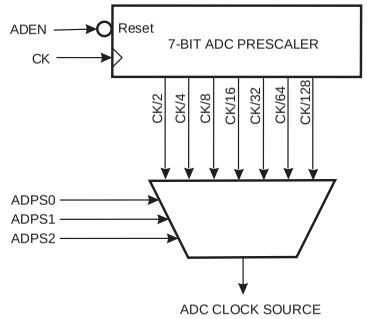
When reading data, ADCL must be read first. Once ADCL is read, ADC access to data registers is blocked. This mea that if ADCL has been read, and a conversion completes before ADCH is read, none of the registers are updated and t result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL registers is re-enabled.

The ADC has its own interrupt, ADIF, which can be triggered when a conversion completes. When ADC access to the da registers is prohibited between reading of ADCL and ADCH, the interrupt will trigger even if the result is lost.

grifo® ITALIAN TECHNOLOGY

Prescaling

Figure 46. ADC Prescaler



The ADC contains a prescaler, which divides the system clock to an acceptable ADC clock frequency. The ADC accep input clock frequencies in the range 50 - 200 kHz. Applying a higher input frequency will result in a poorer accuracy, s "ADC DC Characteristics" on page 72.

The ADPS0 - ADPS2 bits in ADCSR are used to generate a proper ADC clock input frequency from any XTAL frequen above 100 kHz. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCS The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following falling edge of t ADC clock cycle. The actual sample-and-hold takes place one ADC clock cycle after the start of the conversion. The res is ready and written to the ADC Result Register after 13 cycles. The ADC needs 2 more clock cycles before a new conve sion can be started. If ADSC is set high in this period, the ADC will start the new conversion immediately. For a summary conversion times, see Table 27.

Figure 47. ADC timing diagram, first conversion

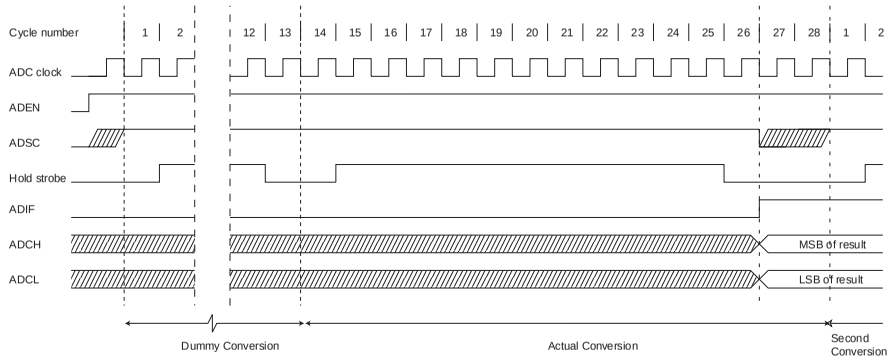
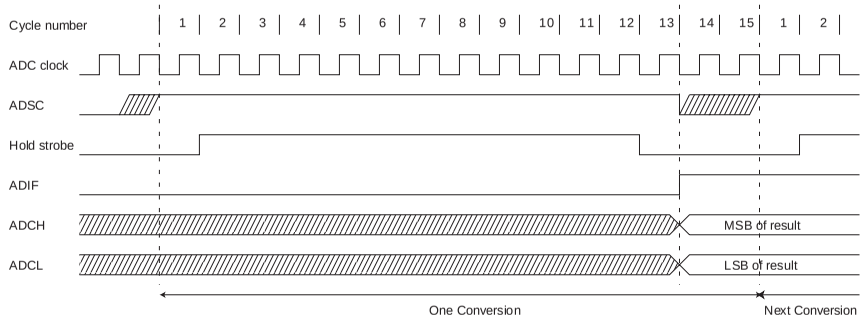


Table 27. ADC Conversion Time

Condition	Sample Cycle Number	Result Ready (cycle number)	Total Conversion Time (cycles)	Total Conversion Time (µs)
1st Conversion	14	26	28	140 - 560
Single Conversion	1	13	15	75 - 300

Figure 48. ADC Timing Diagram



ADC Noise Canceler Function

The ADC features a noise canceler that enables conversion during idle mode to reduce noise induced from the CPU co To make use of this feature, the following procedure should be used:

1. Turn off the ADC by clearing ADEN.
2. Turn on the ADC and simultaneously start a conversion by setting ADEN and ADSC. This starts a dummy conversion that will be followed by a valid conversion.
3. Within 14 ADC clock cycles, enter idle mode.
4. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the MCU and execute the ADC conversion complete interrupt routine.

ADC Multiplexer Select Register - ADMUX

Bit	7	6	5	4	3	2	1	0	
$07 ($27)	-	-	-	-	-	MUX2	MUX1	MUX0	ADMUX
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..3 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always read as zero.

- **Bits 2..0 - MUX2..MUX0: Analog Channel Select Bits 2-0**

The value of these three bits selects which analog input 7-0 is connected to the ADC.

ADC Control and Status Register - ADCSR

Bit	7	6	5	4	3	2	1	0	
$06 ($26)	ADEN	ADSC	-	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - ADEN: ADC Enable**

Writing a logical '1' to this bit enables the ADC. By clearing this bit to zero, the ADC is turned off. Turning the ADC off wh a conversion is in progress, will terminate this conversion.

- **Bit 6 - ADSC: ADC Start Conversion**

A logical '1' must be written to this bit to start each conversion. The first time ADSC has been written after the ADC h been enabled, or if ADSC is written at the same time as the ADC is enabled, a dummy conversion will precede the initiat conversion. This dummy conversion performs initialization of the ADC.

ADSC remains high during the conversion. ADSC goes low after the conversion is complete, but before the result is writt to the ADC Data Registers. This allows a new conversion to be initiated before the current conversion is complete. The ne conversion will then start immediately after the current conversion completes. When a dummy conversion precedes a re conversion, ADSC will stay high until the real conversion completes.

Writing a zero to this bit has no effect.

- **Bit 5 - Res: Reserved Bit**

This bit is reserved in the ATmega603/103. **Warning**: When writing ADCSR, a logical "0" must be written to this bit.

- **Bit 4 -ADIF: ADC Interrupt Flag**

This bit is set (one) when an ADC conversion is complete and the the result is written to the ADC Data Registers a updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set (one). ADIF cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing logical one to the flag. Beware that if doing a read-modify-write on ADCSR, a pending interrupt can be disabled. This al applies if the SBI and CBI instructions are used.

- **Bit 3 - ADIE: ADC Interrupt Enable**

When this bit is set (one) and the I-bit in SREG is set (one), the ADC Conversion Complete Interrupt is activated.

- **Bits 2..0 - ADPS2..ADPS0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 28. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	Invalid
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

ADC Data Register - ADCL and ADCH

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	-	-	-	-	-	-	ADC9	ADC8	ADCH
$04 ($24)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. It is essential that both registers are rea and that ADCL is read before ADCH.

ADC Noise Canceling Techniques

Digital circuitry inside and outside the ATmega603/103 generates EMI which might affect the accuracy of analog measu ments. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the ATmega603/103 and all analog components in the application should have a separate ana- log ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
3. The AV_{CC} pin on the ATmega603/103 should have its own decoupling capacitor as shown in Figure 49.
4. Use the ADC noise canceler function to reduce induced noise from the CPU.
5. If some Port F pins are used as digital inputs, it is essential that these do not switch while a conversion is in progress.

Figure 49. ADC Power Connections

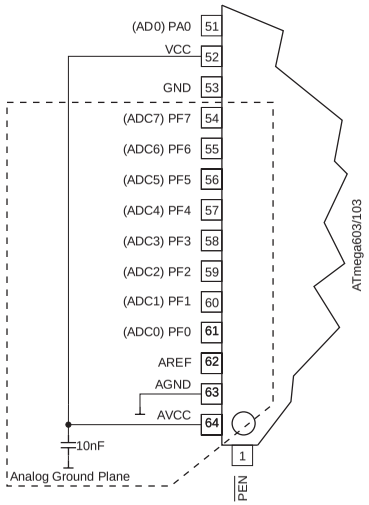
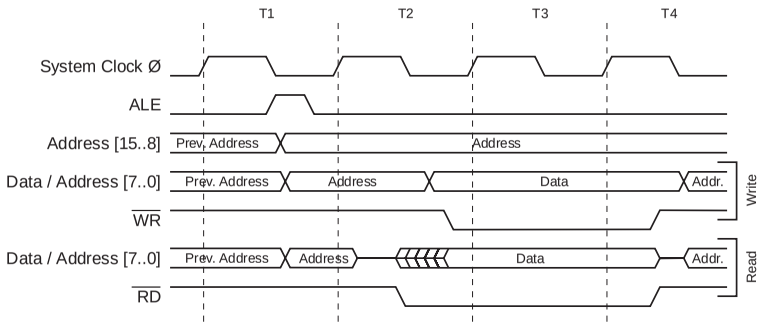


Figure 52. External SRAM Access Cycle with wait state



I/O-Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the dire tion of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and C instructions. The same applies for changing drive value (if configured as output) or enabling/disabling of pull-up resistors configured as input).

Port A

Port A is an 8-bit bi-directional I/O port with internal pull-ups.

Three I/O memory address locations are allocated for Port A, one each for the Data Register - PORTA, $1B($3B), Da Direction Register - DDRA, $1A($3A) and the Port A Input Pins - PINA, $19($39). The Port A Input Pins address is re only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20mA and thus drive LE displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if t internal pull-up resistors are activated.

The Port A pins have alternate functions related to the optional external data SRAM. Port A can be configured to be t multiplexed low-order address/data bus during accesses to the byte.

When Port A is set to the alternate function by the SRE - External SRAM Enable - bit in the MCUCR - MCU Control Reg ter, the alternate settings override the data direction register.

Port A Data Register - PORTA

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port A Data Direction Register - DDRA

Bit	7	6	5	4	3	2	1	0	
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port A Input Pins Address - PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port A Input Pins address - PINA - is not a register, and this address enables access to the physical value on each P A pin. When reading PORTA the Port A Data Latch is read, and when reading PINA, the logical values present on the pi are read.

Port A as General Digital I/O

All 8 pins in Port A have equal functionality when used as digital I/O pins.

PAn, General I/O pin: The DDAn bit in the DDRA register selects the direction of this pin, if DDAn is set (one), PAn is cc figured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the p configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, PORTAn has to l cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becom active, even if the clock is not running.

Table 29. DDAn Effects on Port A Pins

DDAn	PORTAn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

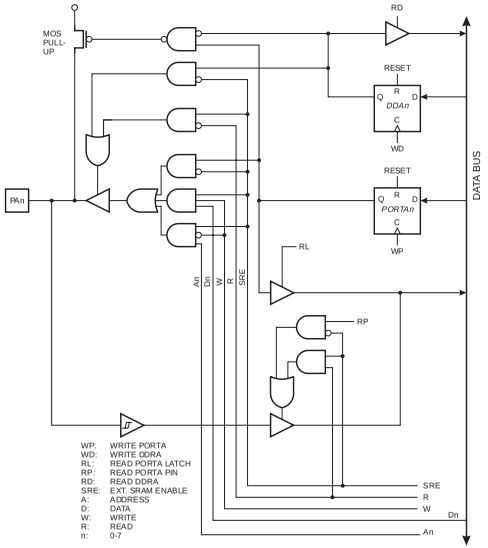
Note: n: 7,6...0, pin number

Port A Schematics

Note that all port pins are synchronized. The synchronization latch is however, not shown in the figure.

GPC® AM4 Rel. 3.00
Pagina B-33
ITALIAN TECHNOLOGY grifo®

Figure 53. Port A Schematic Diagrams (Pins PA0 - PA7)



Port B

Port B is an 8-bit bi-directional I/O port with internal pull-ups.

Three I/O memory address locations are allocated for Port B, one each for the Data Register - PORTB, $18($38), Da Direction Register - DDRB, $17($37) and the Port B Input Pins - PINB, $16($36). The Port B Input Pins address is re only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20mA and thus drive LED d plays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if tl internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in the following table:

Table 30. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	$\overline{SS}$ (SPI Slave Select input)
PB1	SCK (SPI Bus Serial Clock)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB4	OC0/PWM0 (Output Compare and PWM Output for Timer/Counter0)
PB5	OC1A/PWM1A (Output Compare and PWM Output A for Timer/Counter1)
PB6	OC1B/PWM1B (Output Compare and PWM Output B for Timer/Counter1)
PB7	OC2/PWM2 (Output Compare and PWM Output for Timer/Counter2

When the pins are used for the alternate function the DDRB and PORTB register have to be set according to the alterna function description.

Port B Data Register - PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port B Data Direction Register - DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port B Input Pins Address - PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each P B pin. When reading PORTB, the Port B Data Latch is read, and when reading PINB, the logical values present on the pi are read.

Port B as General Digital I/O

All 8 pins in port B have equal functionality when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is cc figured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn is set (one) when the p configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PORTBn has to l cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becom active, even if the clock is not running.

abaco bus grifo® ITALIAN TECHNOLOGY

Table 31. DDBn Effects on Port B Pins

DDBn	PORTBn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PBn will source current if ext. pulled low
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

Note: n: 7,6...0, pin number

Alternate Functions of Port B

The alternate pin configuration is as follows:

- **OC2/PWM2, Bit 7**

OC2/PWM2, Output Compare output for Timer/Counter2 or PWM output when Timer/Counter2 is in PWM Mode. The p has to be configured as an output to serve this function.

- **OC1B/PWM1B, Bit 6**

OC1B/PWM1B, Output Compare output B for Timer/Counter1 or PWM output B when Timer/Counter1 is in PWM Mod The pin has to be configured as an output to serve this function.

- **OC1A/PWM1A, Bit 5**

OC1A/PWM1A, Output Compare output A for Timer/Counter1 or PWM output A when Timer/Counter1 is in PWM Mod The pin has to be configured as an output to serve this function.

- **OC0/PWM0, Bit 4**

OC0/PWM0, Output Compare output for Timer/Counter0 or PWM output when Timer/Counter0 is in PWM Mode. The p has to be configured as an output to serve this function.

- **MISO - Port B, Bit 3**

MISO: Master data input, slave data output pin for SPI channel. When the SPI is enabled as a master, this pin is configur as an input regardless of the setting of DDB3. When the SPI is enabled as a slave, the data direction of this pin is controll by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit. See the description the SPI port for further details.

- **MOSI - Port B, Bit 2**

MOSI: SPI Master data output, slave data input for SPI channel. When the SPI is enabled as a slave, this pin is configur as an input regardless of the setting of DDB2. When the SPI is enabled as a master, the data direction of this pin is co trolled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See tl description of the SPI port for further details.

- **SCK - Port B, Bit 1**

SCK: Master clock output, slave clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configur as an input regardless of the setting of DDB1. When the SPI is enabled as a master, the data direction of this pin is co trolled by DDB1. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB1 bit. See tl description of the SPI port for further details.

- **$\overline{SS}$ - Port B, Bit 0**

$\overline{SS}$: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setti of DDB0. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data dire tion of this pin is controlled by DDB0. When the pin is forced to be an input, the pull-up can still be controlled by tl PORTB0 bit. See the description of the SPI port for further details.

Port B Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

Figure 54. Port B Schematic Diagram (Pin PB0)

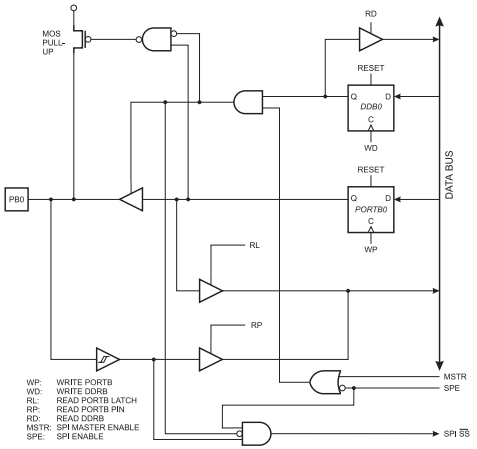


Figure 55. Port B Schematic Diagram (Pin PB1)

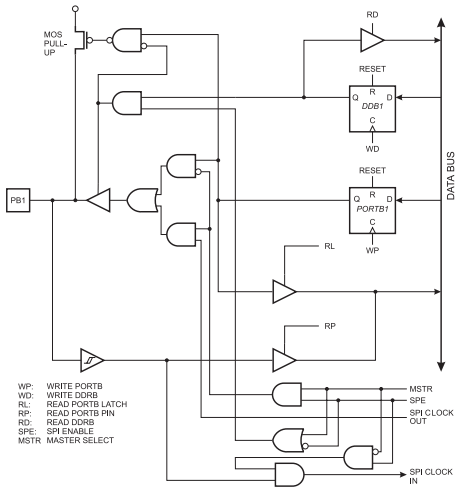


Figure 56. Port B Schematic Diagram (Pin PB2)

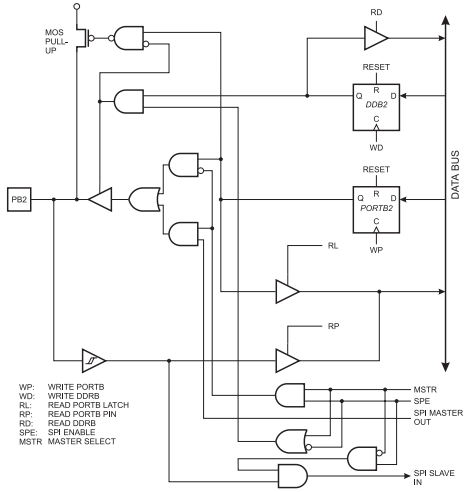


Figure 57. Port B Schematic Diagram (Pin PB3)

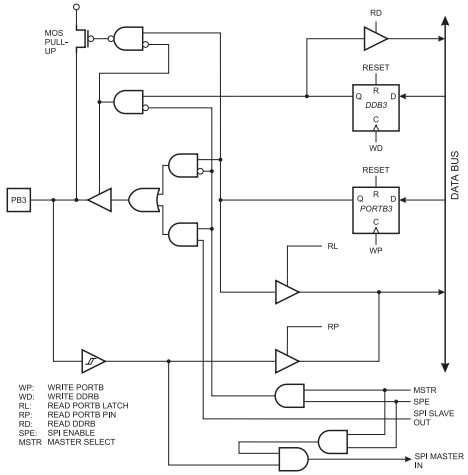


Figure 58. Port B Schematic Diagram (Pin PB4)

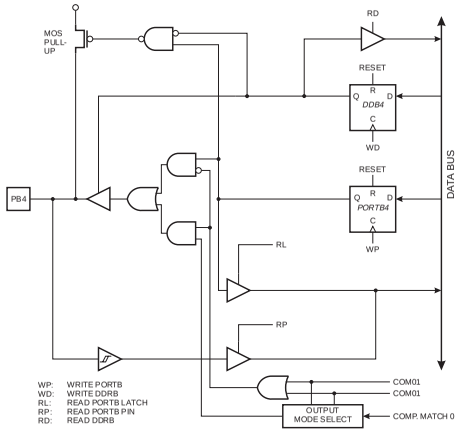
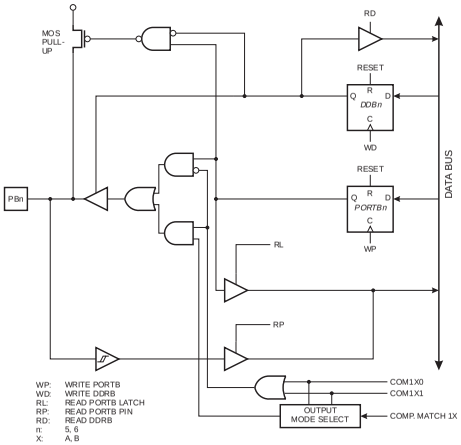
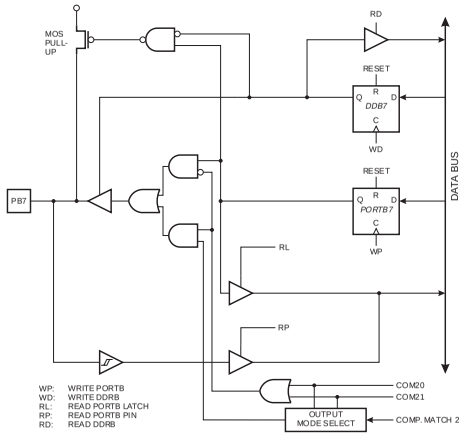


Figure 59. Port B Schematic Diagram (Pins PB5 and PB6)



grifo®
ITALIAN TECHNOLOGY

Figure 60. Port B Schematic Diagram (Pin PB7)



Port C

PORT C is an 8-bit Output port.

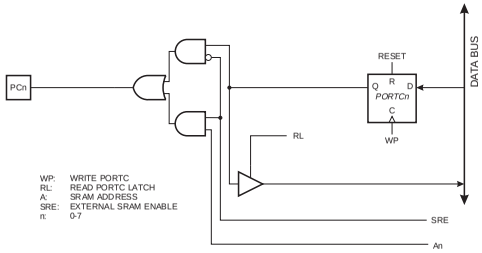
The Port C pins have alternate functions related to the optional external data SRAM. When using the device with extern SRAM, Port C outputs the high-order address byte during accesses to external data memory. When a reset conditi becomes active, the port pins are not tristated, but the pins will assume their initial value after two stable clock cycles.

The Port C Data Register - PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port C Schematics

Figure 61. Port C Schematic Diagram (Pins PC0 - PC7)



Port D

Port D is an 8 bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for the Port D, one each for the Data Register - PORTD, $12($32), Da Direction Register - DDRD, $11($31) and the Port D Input Pins - PIND, $10($30). The Port D Input Pins address is re only, while the Data Register and the Data Direction Register are read/write.

The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pu up resistors are activated.

Some Port D pins have alternate functions as shown in the following table:

Table 32. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	$\overline{INT0}$ (External Interrupt0 Input)
PD1	$\overline{INT1}$ (External Interrupt1 Input)
PD2	$\overline{INT2}$ (External Interrupt2 Input)
PD3	$\overline{INT3}$ (External Interrupt3 Input)
PD4	IC1 (Timer/Counter1 Input Capture Trigger)
PD6	T1 (Timer/Counter1 Clock Input)
PD7	T2 (Timer/Counter2 Clock Input)

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alterna function description.

Port D Data Register - PORTD

Bit	7	6	5	4	3	2	1	0	
$12	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port D Data Direction Register - DDRD

Bit	7	6	5	4	3	2	1	0	
$11	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port D Input Pins Address - PIND

Bit	7	6	5	4	3	2	1	0	
$10	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on ea Port D pin. When reading PORTD, the Port D Data Latch is read, and when reading PIND, the logical values present on tl pins are read.

Port D as general digital I/O

PDn, General I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is co figured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configur as an input pin the MOS pull up resistor is activated. To switch the pull up resistor off the PDn has to be cleared (zero) the pin has to be configured as an output pin.The port pins are tri-stated when a reset condition becomes active, even if tl clock is not running.

Table 33. DDDn Bits on Port D Pins

DDDn	PORTDn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

Note: n: 7,6...0, pin number

Alternate Functions of Port D

$\overline{INT0}$ .. $\overline{INT3}$ - Port D, Bits 0..3

External Interrupt sources 0 - 3. The PD0 - PD3 pins can serve as external active low interrupt sources to the MCU. Tl internal pull up MOS resistors can be activated as described above. See the interrupt description for further details, a how to enable the sources.

IC1 - Port D, Bit 4

IC1 - Input Capture pin for Timer/Counter1. When a positive or negative (selectable) edge is applied to this pin, the co tents of Timer/Counter1 is transferred to the Timer/Counter1 Input Capture Register. The pin has to be configured as input to serve this function. See the Timer/Counter1 description on how to operate this function. The internal pull up MC resistor can be activated as described above.

T1 - Port D, Bit 6

T1, Timer/Counter1 counter source. See the timer description for further details.

T2 - Port D, Bit 7

T2, Timer/Counter2 counter source. See the timer description for further details.

Port D Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

Figure 62. Port D Schematic Diagram (Pins PD0, PD1, PD2 and PD3)

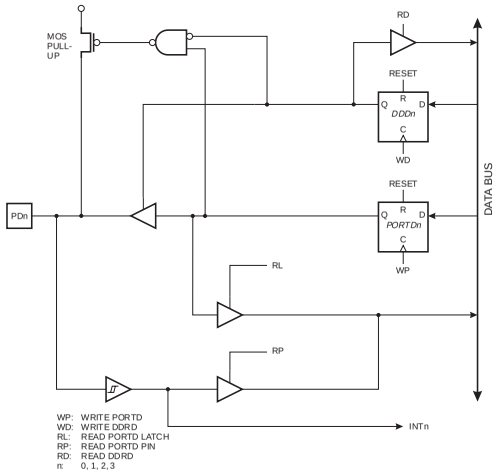


Figure 63. Port D Schematic Diagram (Pin PD4)

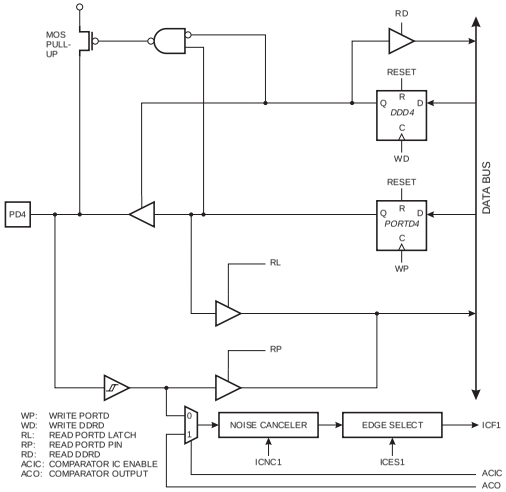


Figure 64. Port D Schematic Diagram (Pin PD5)

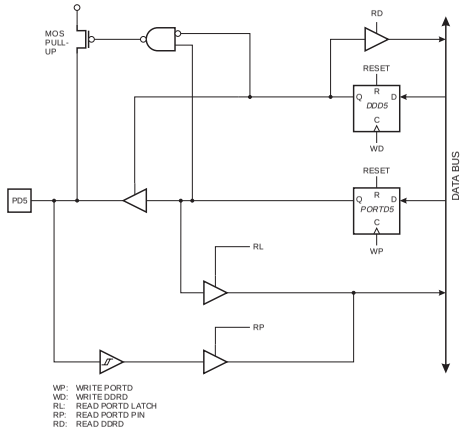
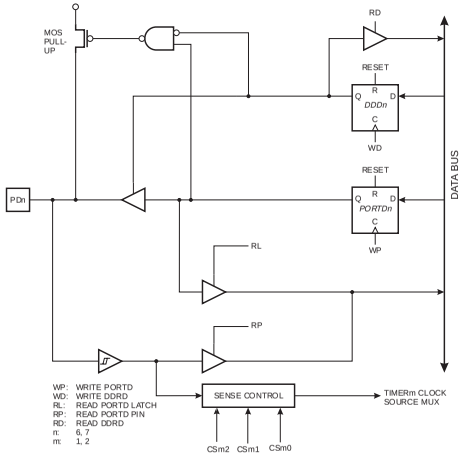


Figure 65. Port D Schematic Diagram (Pins PD6 and PD7)



Port E

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for the Port E, one each for the Data Register - PORTE, $03($23), Da Direction Register - DDRE, $02($22) and the Port E Input Pins - PINE, $01($21). The Port E Input Pins address is re only, while the Data Register and the Data Direction Register are read/write.

The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pu up resistors are activated.

All Port E pins have alternate functions as shown in the following table:

Table 34. Port E Pins Alternate Functions

Port Pin	Alternate Function
PE0	PDI/RXD (Programming Data Input or UART Receive Pin)
PE1	PDO/TXD (Programming Data Output or UART Transmit Pin)
PE2	AC+ (Analog Comparator Positive Input)
PE3	AC- (Analog Comparator Negative Input)
PE4	INT4 (External Interrupt4 Input)
PE5	INT5 (External Interrupt5 Input)
PE6	INT6 (External Interrupt6 Input)
PE7	INT7 (External Interrupt7 Input)

When the pins are used for the alternate function the DDRE and PORTE register has to be set according to the alterna function description.

Port E Data Register - PORTE

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	PORTE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port E Data Direction Register - DDRE

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	DDRE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port E Input Pins Address - PINE

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	PINE
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port E Input Pins address - PINE - is not a register, and this address enables access to the physical value on each P E pin. When reading PORTE, the Port E Data Latch is read, and when reading PINE, the logical values present on the pi are read.

Port E as general digital I/O

PEn, General I/O pin: The DDEn bit in the DDRE register selects the direction of this pin. If DDEn is set (one), PEn is co figured as an output pin. If DDEn is cleared (zero), PEn is configured as an input pin. If PEn is set (one) when configured an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off the PEn has to be cleared (zero) or t pin has to be configured as an output pin.The port pins are tri-stated when a reset condition becomes active, even if t clock is not running.

Table 35. DDEn Bits on Port E Pins

DDEn	PORTEn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

Note: n: 7,6...0, pin number

Alternate Functions OF Port E

PDI/RXD - Port E, Bit 0

PDI, Serial Programming Data Input. During Serial Program Downloading, this pin is used as data input line for t ATmega603/103.

RXD, UART Receive Pin. Receive Data (Data input pin for the UART). When the UART receiver is enabled this pin is co figured as an input regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical one PORTD0 will turn on the internal pull-up.

PDO/TXD - Port E, Bit 1

PDO, Serial Programming Data Output. During Serial Program Downloading, this pin is used as data output line for t ATmega603/103.

TXD, UART Transmit Pin.

AC+ - Port E, Bit 2

AC+ - Analog Comparator Positive Input. This pin is directly connected to the positive input of the analog comparator.

AC- - Port E, Bit 3

AC- - Analog Comparator Negative Input. This pin is directly connected to the negative input of the analog comparator.

INT4 .. INT7 - Port E, Bit 4-7

INT4 .. INT7 - External Interrupt sources 4 - 7: The PE4 - PE7 pins can serve as external interrupt sources to the MC Interrupts can be triggered by low level or positive or negative edge on these pins. The internal pull up MOS resistors c be activated as described above. See the interrupt description for further details, and how to enable the sources.

Port E Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

Figure 66. Port E Schematic Diagram, Pin PE0

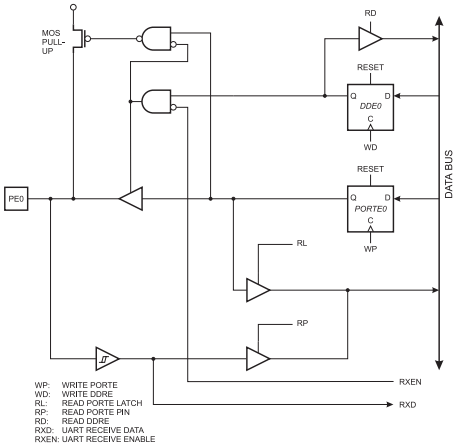
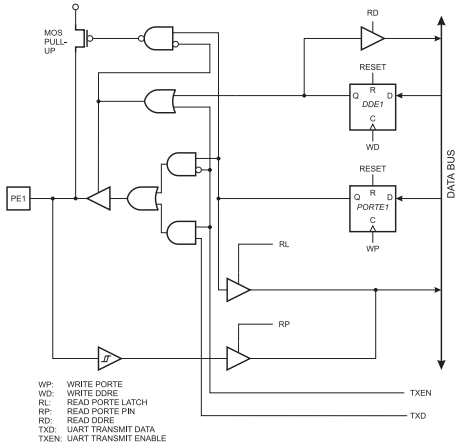


Figure 67. Port E Schematic Diagram (Pin PE1)



GPC® AM4 Rel. 3.00

grifo® ITALIAN TECHNOLOGY

Figure 68. Port E Schematic Diagram (Pin PE2)

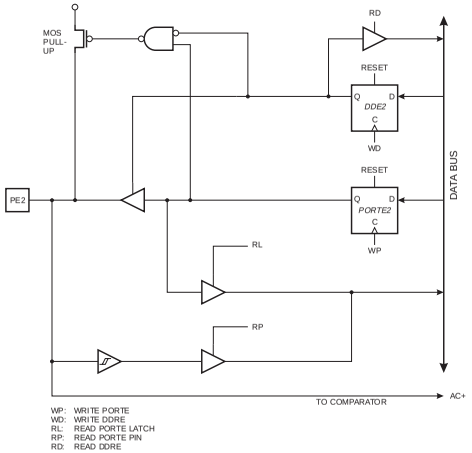


Figure 69. Port E Schematic Diagram (Pin PE3)

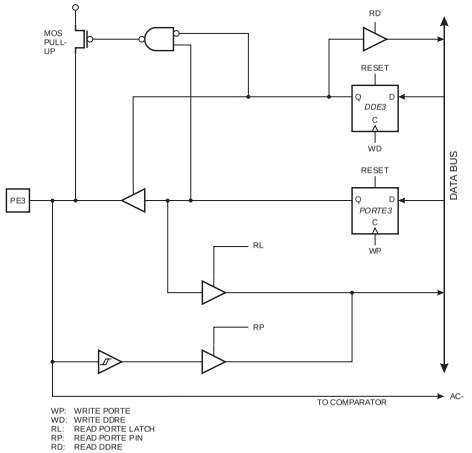
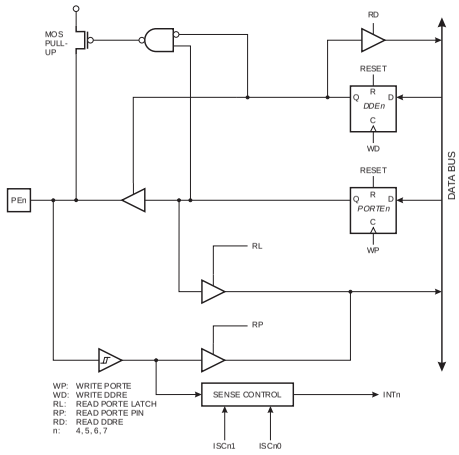


Figure 70. Port E Schematic Diagram (Pins PE4, PE5, PE6 and PE7)



Port F

Port F is an 8-bit input port.

One I/O memory location is allocated for Port F, the Port F Input Pins - PINF, $00 ($20).

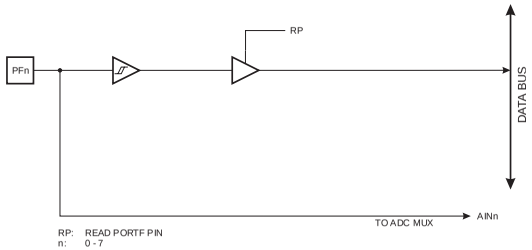
All Port F pins are connected to the analog multiplexer which further is connected to the A/D converter. The digital inp function of Port F can be used together with the A/D converter, allowing the user to use some pins of Port F and digi inputs and other as analog inputs, at the same time.

Port F Input Pins Address - PINF

Bit	7	6	5	4	3	2	1	0	
$00 ($20)	**PINF7**	**PINF6**	**PINF5**	**PINF4**	**PINF3**	**PINF2**	**PINF1**	**PINF0**	**PINF**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port F Input Pins address - PINF - is not a register, and this address enables access to the physical value on each P F pin.

Figure 71. Port F Schematic Diagram (Pins PF7 - PF0)



Memory Programming

Program and Data Memory Lock Bits

The ATmega603/103 MCU provides two Lock bits which can be left unprogrammed ('1') or can be programmed ('0') obtain the additional features listed in Table 36. The Lock bits can only be erased to '1' with the Chip Erase command..

Table 36. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
Mode	LB1	LB2	
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bi

Fuse Bits

The ATmega603/103 has four Fuse bits, SPIEN, SUT1..0, and EESAVE.

- When the SPIEN Fuse is programmed ('0'), Serial Program and Data Downloading is enabled. Default value is programmed ('0'). The SPIEN Fuse is not accessible in serial programming mode.
- When EESAVE is programmed, the EEPROM memory is preserved through the Chip Erase cycle. Default value is unprogrammed ('1'). The EESAVE Fuse bit can not be programmed if any of the Lock bits are programmed.
- SUT1..0 Fuses: Determine the MCU start-up time. See Table 6 on page 26 for further details. Default value is unprogrammed ('11'), which gives a nominal start up time of 16 ms.

The status of the Fuse bits is not affected by Chip Erase.

Register Summary

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	page 21
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	page 21
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 21
$3C ($5C)	XDIV	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	page 23
$3B ($5B)	RAMPZ	-	-	-	-	-	-	-	RAMPZ0	page 22
$3A ($5A)	EICR	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	page 31
$39 ($59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	page 30
$38 ($58)	EIFR	INTF7	INTF6	INTF5	INTF4	-	-	-	-	page 30
$37 ($57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	page 31
$36 ($56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	page 32
$35 ($55)	MCUCR	SRE	SRW	SE	SM1	SM0	-	-	-	page 22
$34 ($54)	MCUSR	-	-	-	-	-	-	EXTRF	PORF	page 29
$33 ($53)	TCCR0	-	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	page 38
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								page 39
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								page 40
$30 ($50)	ASSR	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	page 41
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	page 45
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	page 46
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								page 47
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								page 47
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								page 48
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								page 48
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								page 48
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								page 48
$27 ($47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								page 48
$26 ($46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								page 48
$25 ($45)	TCCR2	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	page 38
$24 ($44)	TCNT2	Timer/Counter2 (8 Bit)								page 39
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								page 40
$21 ($47)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	page 51
$1F ($3F)	EEARH	-	-	-	-	EEAR11	EEAR10	EEAR9	EEAR8	page 52
$1E ($3E)	EEARL	EEPROM Address Register L								page 52
$1D ($3D)	EEDR	EEPROM Data Register								page 53
$1C ($3C)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	page 53
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	page 75
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	page 75
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	page 75
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 77
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	page 77
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 77
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	page 82
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 83
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 84
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 84
$0F ($2F)	SPDR	SPI Data Register								page 58
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	page 58
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 57
$0C ($2C)	UDR	UART I/O Data Register								page 62
$0B ($2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	page 62
$0A ($2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 63
$09 ($29)	UBRR	UART Baud Rate Register								page 65
$08 ($28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 65
$07 ($27)	ADMUX	-	-	-	-	-	MUX2	MUX1	MUX0	page 69
$06 ($26)	ADCSR	ADES	ABSY	ADRF	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 70
$05 ($25)	ADCH	-	-	-	-	-	-	ADC9	ADC8	page 71
$04 ($24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	page 71
$03 ($23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	page 87
$02 ($22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	page 87
$01 ($21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	page 87
$00 ($20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	page 91

Note: For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instr tions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI an SBI instructions work with registers $00 to $1F only.

grifo® ITALIAN TECHNOLOGY

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd - ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
ELPM[1]		Extended Load Program Memory	R0 ← (Z+RAMPZ)	None	3
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watchdog Reset	(see specific descr. for WD timer)	None	1

Note: 1. Not in ATmega603.

APPENDICE C: SCHEMI ELETTRICI

In questa appendice sono disponibili gli schemi elettrici delle interfaccie per la **GPC® AM4** più frequentemente utilizzate. Tutte queste interfaccie possono essere prodotte autonomamente dall'utente mentre solo alcune di esse sono schede **grifo®** standard e possono quindi essere ordinate.

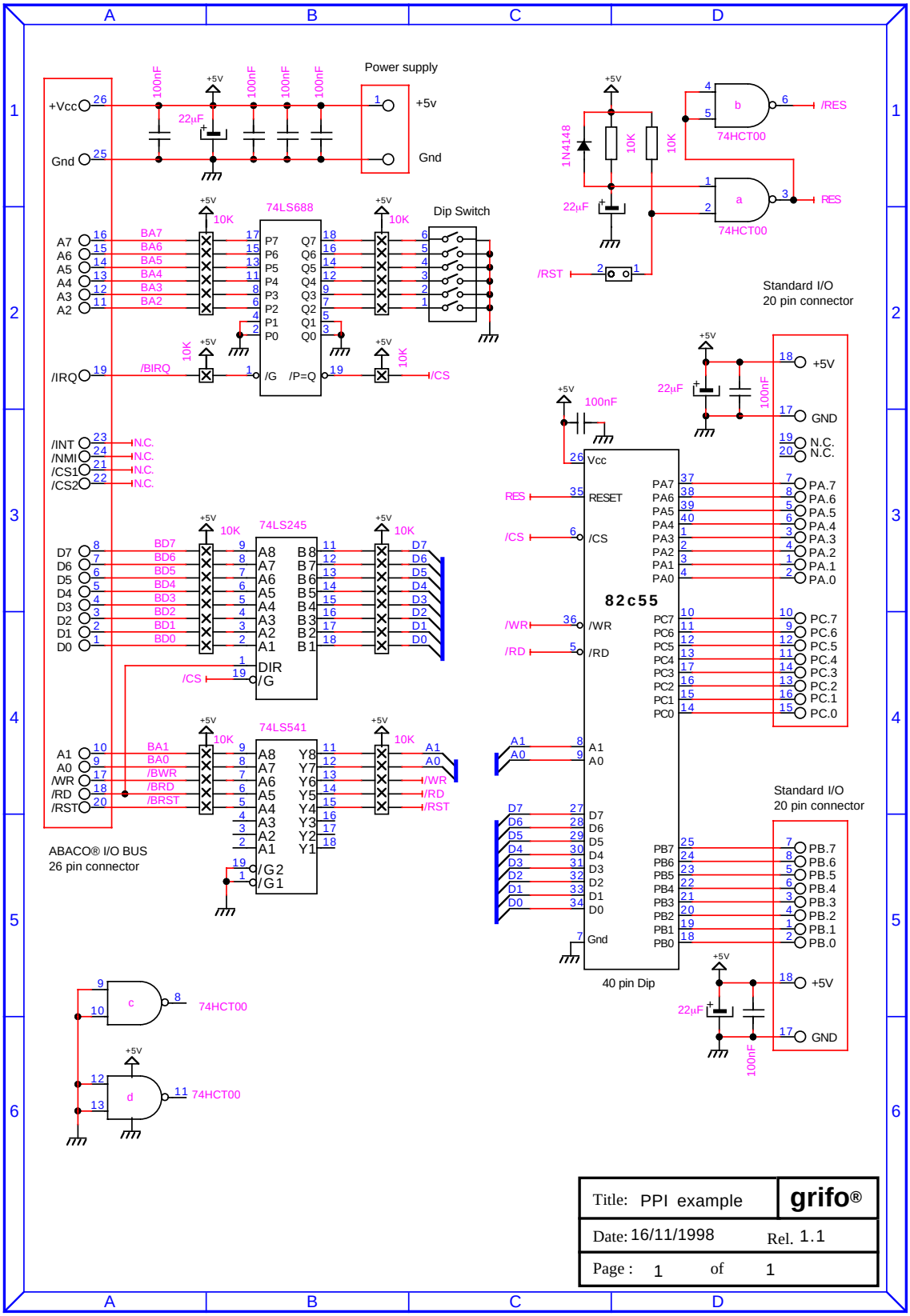


FIGURA C1: SCHEMA ELETTRICO DI ESPANSIONE PPI

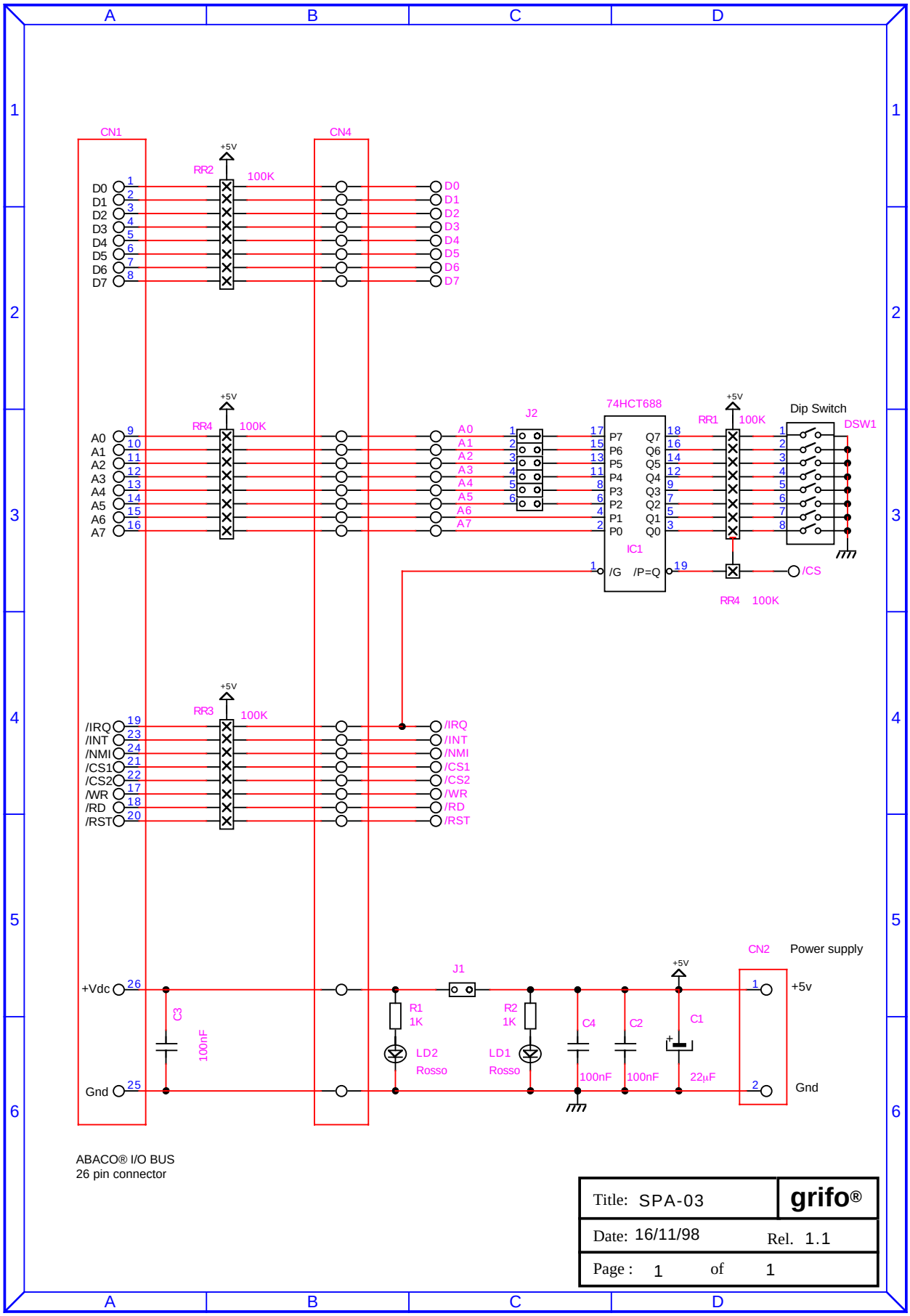


FIGURA C2: SCHEMA ELETTRICO SPA 03

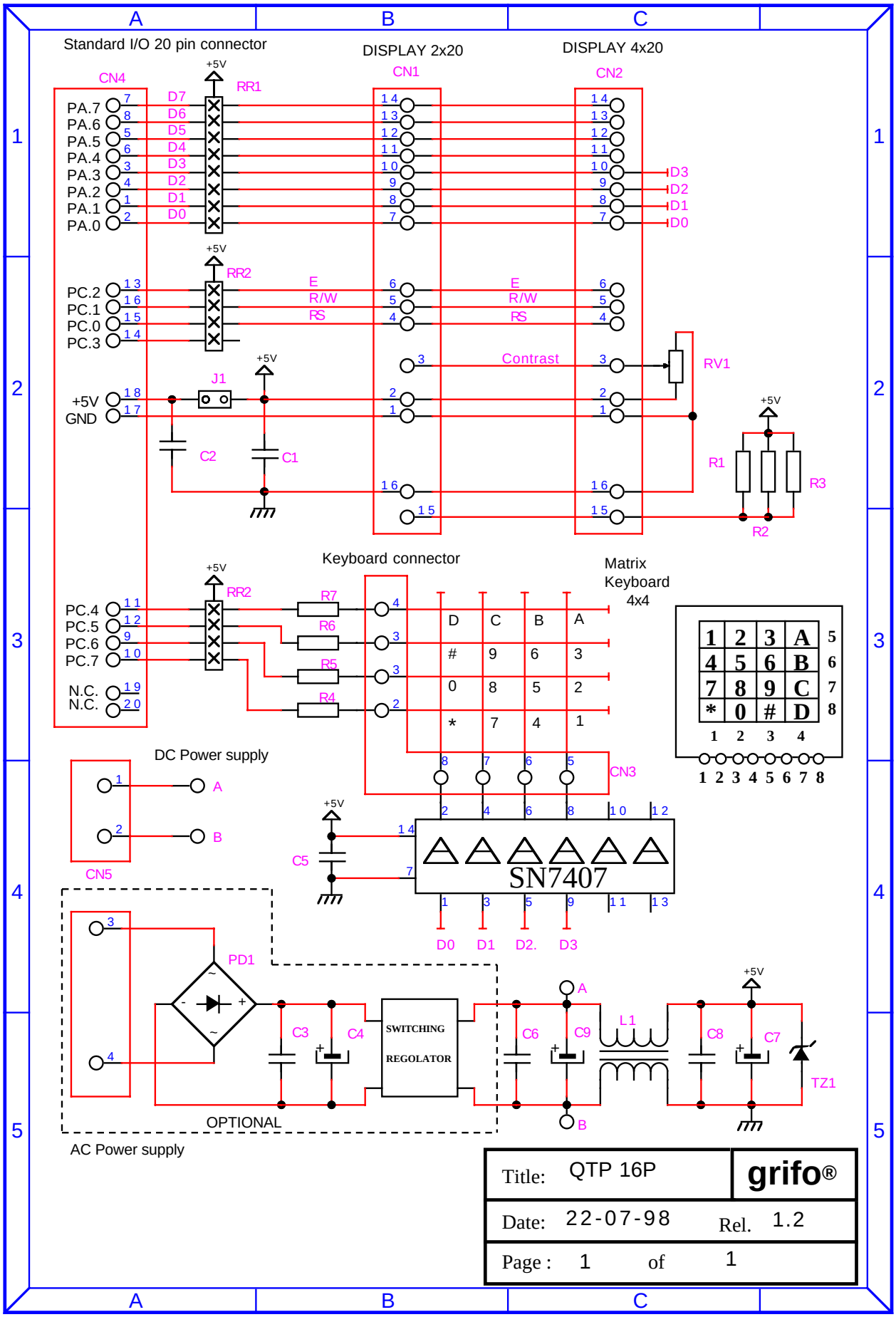


FIGURA C3: SCHEMA ELETTRICO QTP 16P

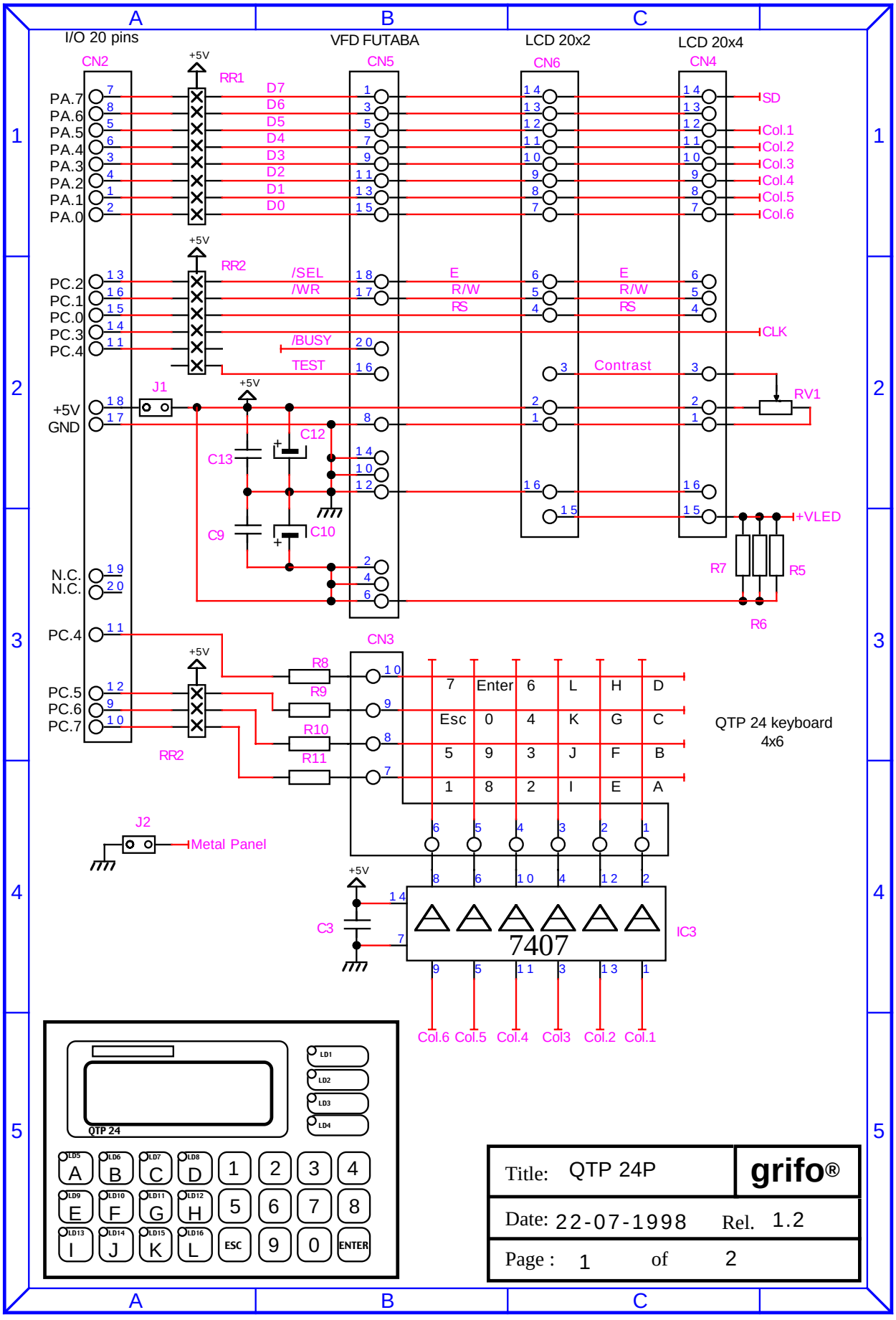


FIGURA C4: SCHEMA ELETTRICO QTP 24P 1/2

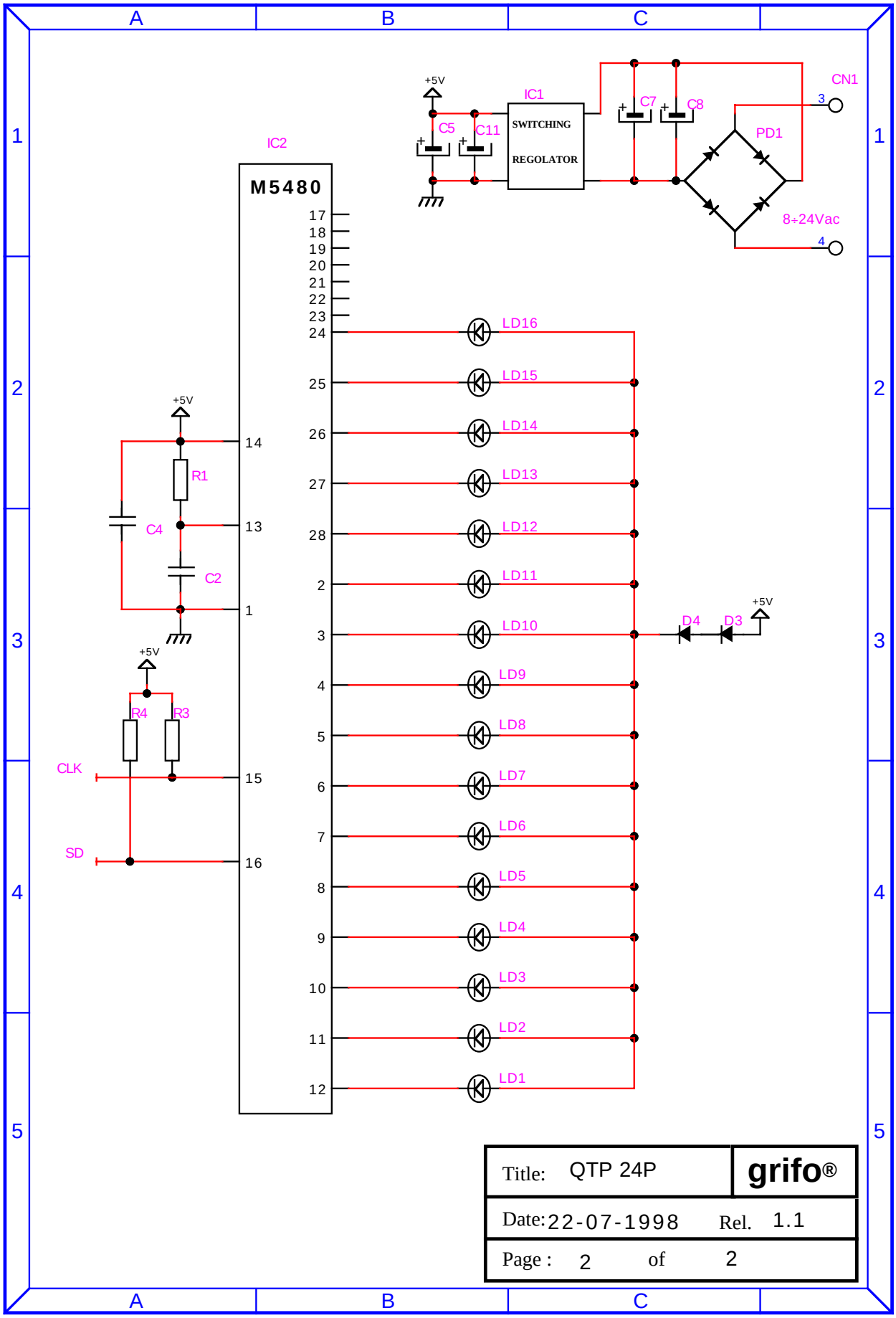


FIGURA C5: SCHEMA ELETTRICO QTP 24P 2/2

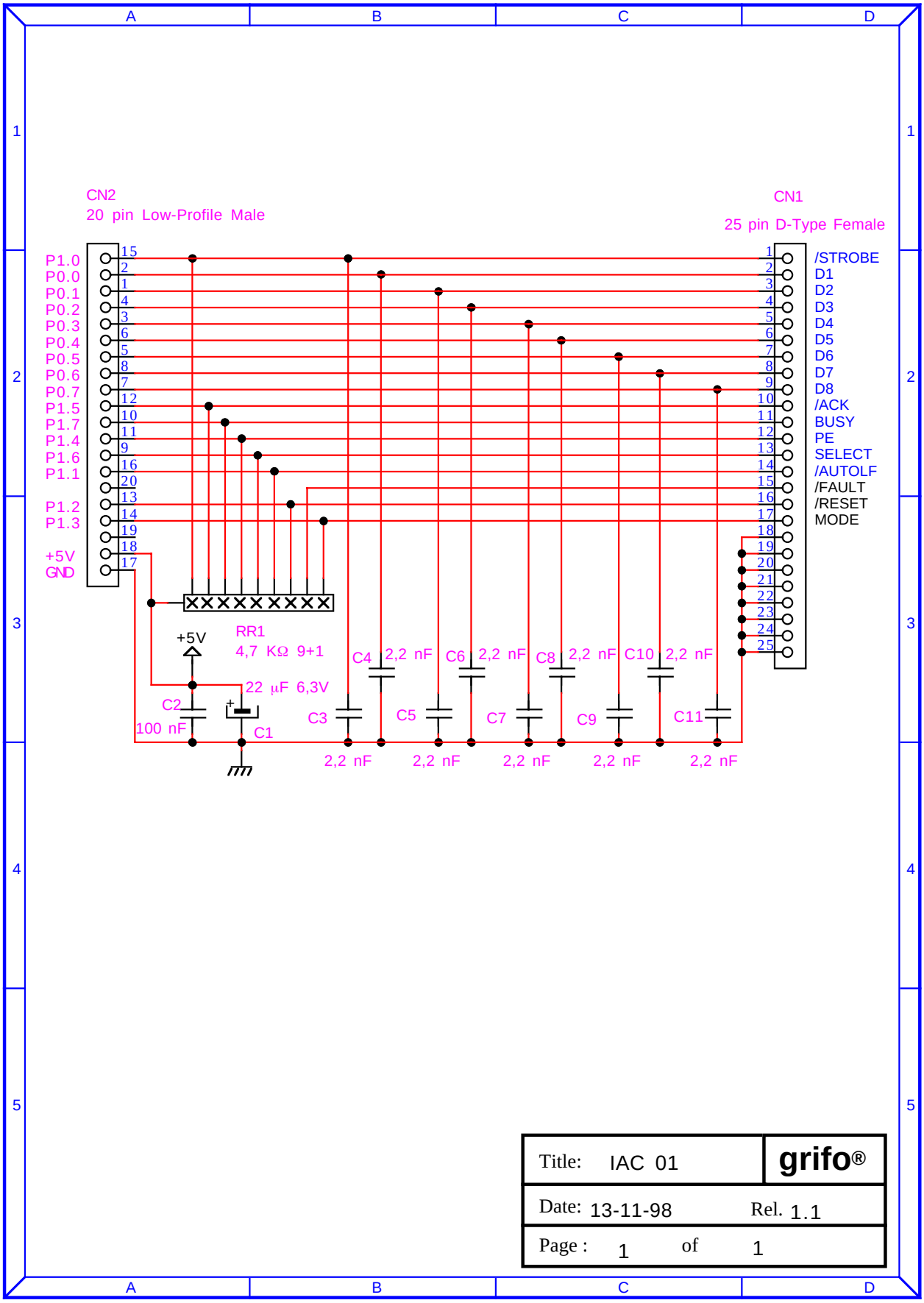
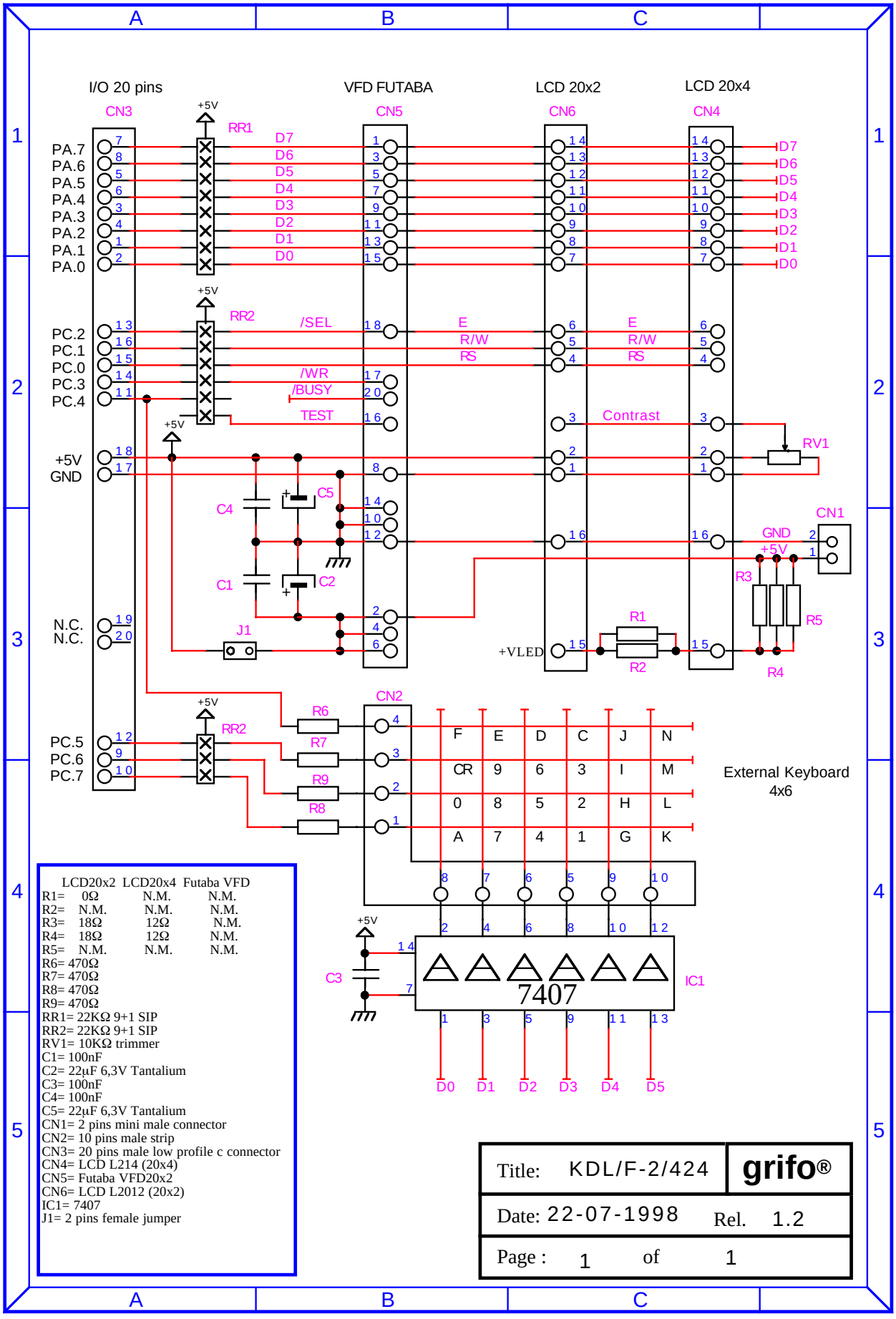


FIGURA C6: SCHEMA ELETTRICO IAC 01



	LCD20x2	LCD20x4	Futaba VFD
R1=	0Ω	N.M.	N.M.
R2=	N.M.	N.M.	N.M.
R3=	18Ω	12Ω	N.M.
R4=	18Ω	12Ω	N.M.
R5=	N.M.	N.M.	N.M.

R6= 470Ω
R7= 470Ω
R8= 470Ω
R9= 470Ω
RR1= 22KΩ 9+1 SIP
RR2= 22KΩ 9+1 SIP
RV1= 10KΩ trimmer
C1= 100nF
C2= 22µF 6,3V Tantalium
C3= 100nF
C4= 100nF
C5= 22µF 6,3V Tantalium
CN1= 2 pins mini male connector
CN2= 10 pins male strip
CN3= 20 pins male low profile c connector
CN4= LCD L214 (20x4)
CN5= Futaba VFD20x2
CN6= LCD L2012 (20x2)
IC1= 7407
J1= 2 pins female jumper

Title:	KDL/F-2/424	grifo®
Date:	22-07-1998	Rel. 1.2
Page :	1 of 1	

FIGURA C7: SCHEMA ELETTRICO KDX X24

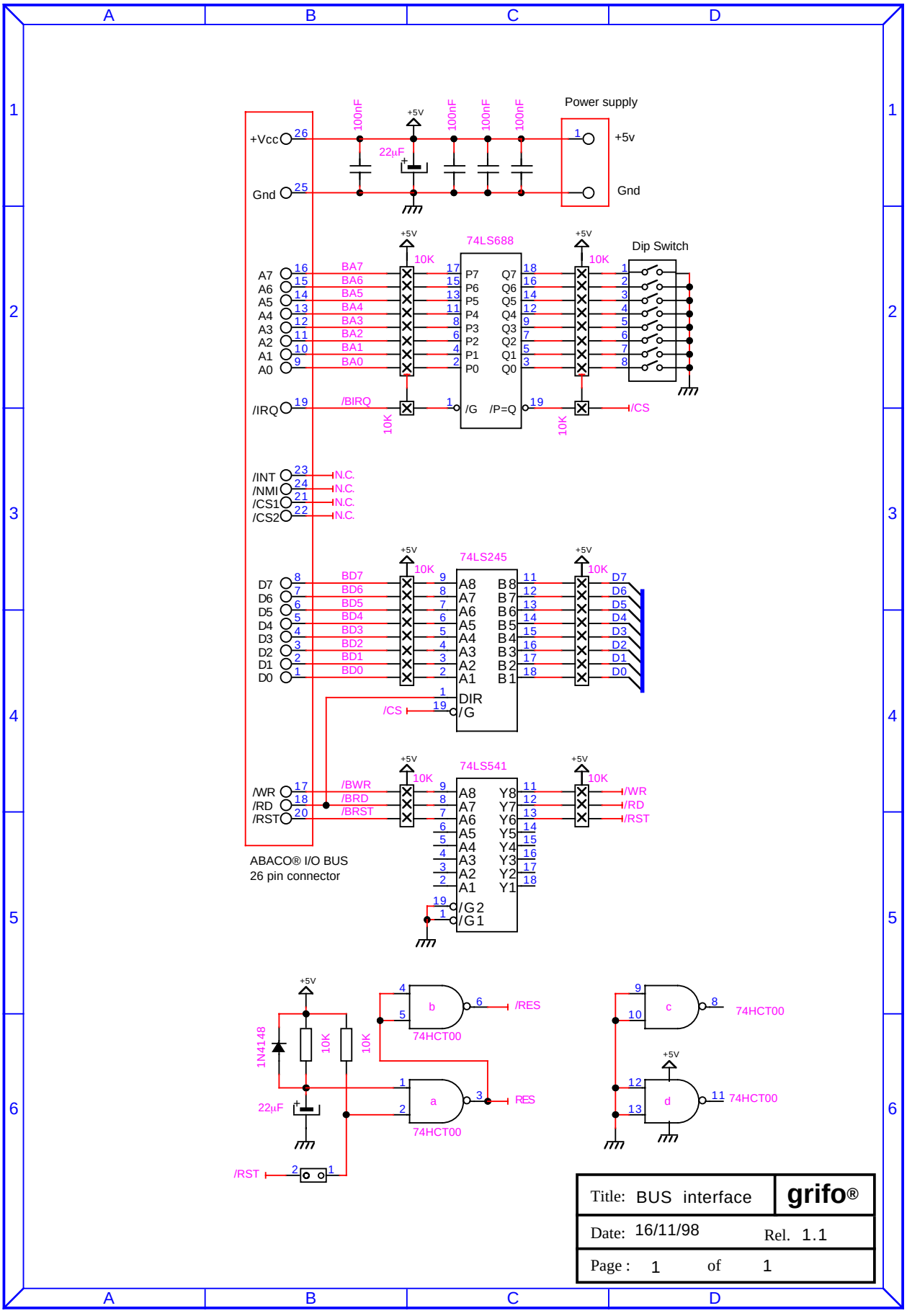


FIGURA C8: SCHEMA ELETTRICO INTERFACCIA BUS

APPENDICE D: INDICE ANALITICO

Simboli

/IORQ **11**, **40**

A

A/D **6**, **8**, **20**, **26**
ABACO® I/O BUS **4**, **8**, **11**, **26**, **31**, **40**, **C-8**
Accesso memorie **42**
Alimentazione **3**
Alimentazione current loop **17**
Anno **45**
Assistenza **1**
ATmega103 **4**, **B-1**
AVR Studio **39**

B

Back up **6**, **8**, **9**, **10**, **31**
Barre Ω **A-1**
BASCOM AVR **39**
Batteria **9**, **10**, **25**, **31**
Bibliografia **50**
BUS ABACO® **4**, **49**
BUS dati **11**
BUS indirizzi **11**

C

Caratteristiche
  elettriche **9**
  fisiche **8**
  generali **8**
CCITT **12**
Clock **3**, **8**
Comparatore analogico **9**, **18**, **27**, **31**
Compilatori **39**
Comunicazione seriale **3**, **12**, **32**
Condensa **8**
Configurazione base **3**, **6**, **9**, **28**, **32**
Connessioni **10**, **49**
Connettori **8**, **10**, **25**, **A-1**
  CN1 **11**
  CN2 **10**
  CN3 **12**
  CN5 **18**
  CN6 **20**
  CN9 **22**

Contatto reset **25**, **36**
Contenitore **1**, **8**, **A-1**
Corrente assorbita **9**
Counter **7**, **18**, **20**
CPU **3**, **4**, **8**, **31**, **40**, **46**, **B-1**
Current loop **3**, **12**, **16**, **26**, **32**

D

Data **45**
DDS MICRO C AVR **39**
DEBUG **44**
Dima **A-1**
Dimensioni **8**, **A-1**
Dip switch **25**, **43**
Disposizione elementi **25**
Disturbi **26**
Driver seriali **33**

E

EEPROM **8**
Errore complessivo A/D **8**
Espandibilità **C-1**
Espansione **4**, **47**
Espansioni I/O **C-1**

F

Filtri **20**
FLASH EPROM **8**, **38**
Forzatura **15**
Frequenza taglio **26**
Frequenze **8**

G

Garanzia **1**
Giorno **45**
Guide DIN **8**, **A-1**

I

I/O ABACO® **18**, **22**, **27**
I/O digitale **7**, **18**, **27**, **34**
IAC 01 **C-6**
ICC AVR **39**
Impedenza **9**, **20**
Indirizzamenti **40**
Informazioni generali **2**

Ingressi analogici **9**, **20**, **26**
Ingressi configurazione **44**
Installazione **10**
Interfacciamento **26**, **27**
Interfaccie **C-1**
Interfaccie operatore **27**, **47**, **C-3**, **C-4**, **C-7**
Interrupts **11**, **18**, **31**, **46**
Introduzione **1**
ISP **7**, **22**, **38**
Istruzioni **4**

J

Jumper configurazione **44**
Jumpers **28**
  2 vie **30**
  3 vie **30**
  a stagno **31**
  disposizione **29**

K

KDx x24 **C-7**

L

LED **25**, **28**

M

Manutenzione **1**
Mappaggio **40**
  ABACO® I/O BUS **40**
  memorie **41**
  periferiche **43**
  risorse **40**
Memorie **6**, **8**, **25**, **41**
Memory card **27**, **48**
Mese **45**
Minuti **45**
Montaggio **1**, **A-1**
Mother board **4**, **48**

O

Opzioni **3**, **32**, **A-1**
Ora **45**

P

Periferiche **44**
Periferiche interne **B-1**
Peso **8**
Piante componenti **37**
Piggy back **A-1**
Pin multifunzione **19**, **34**
Processore **B-1**
PWM **18**, **20**

Q

QTP 16P **C-3**
QTP 24P **C-4**

R

RAM **8**
Real Time Clock **43**, **45**
Registri esterni **43**, **44**
Registri interni **B-8**
Reset **11**, **35**, **36**
Rete current loop **17**
Rete RS 485 **15**
Rete terminazione **9**, **15**, **34**
Retrigger **36**
Risoluzione A/D **8**
Risorse della scheda **8**
RS 232 **3**, **12**, **14**, **26**, **32**
RS 422 **3**, **9**, **12**, **14**, **26**, **32**
RS 485 **3**, **9**, **12**, **14**, **26**, **34**
RTC **3**, **6**, **8**, **31**, **45**
RUN **44**

S

Schede esterne **47**
Schema
  A/D **21**
  Port **19**
Schema a blocchi **5**
Schemi elettrici **C-1**
Secondi **45**
Segnalazioni visive **28**
Segnali controllo **11**, **40**, **44**
Serigrafie **37**
Settimana **45**
Sicurezza **1**
Software **39**

SPA 03 **C-2**
Specifiche tecniche **8**
SPI **18**
Stampante **27**, **C-6**

T

Tarature **26**
Temperatura **8**
Tempo conversione A/D **8**
Tempo d'accesso **8**
Tempo intervento **8**, **36**
Tensione di alimentazione **9**
Timer **7**, **18**
TransZorb™ **3**
Trimmer **25**, **26**
TTL **26**

U

Umidità **8**

V

Versione scheda **1**
Vibrazioni **26**
Vref **26**

W

Watch dog **6**, **8**, **36**